Charactersing effect of anaemia on mortality in severe malaria

Contents

Background 1

Imputation of missing variables 1

Exploratory analysis 3

Predictive value of anaemia on death adjusting for confounders 5

More complex GAM model 7

Model comparison 9

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

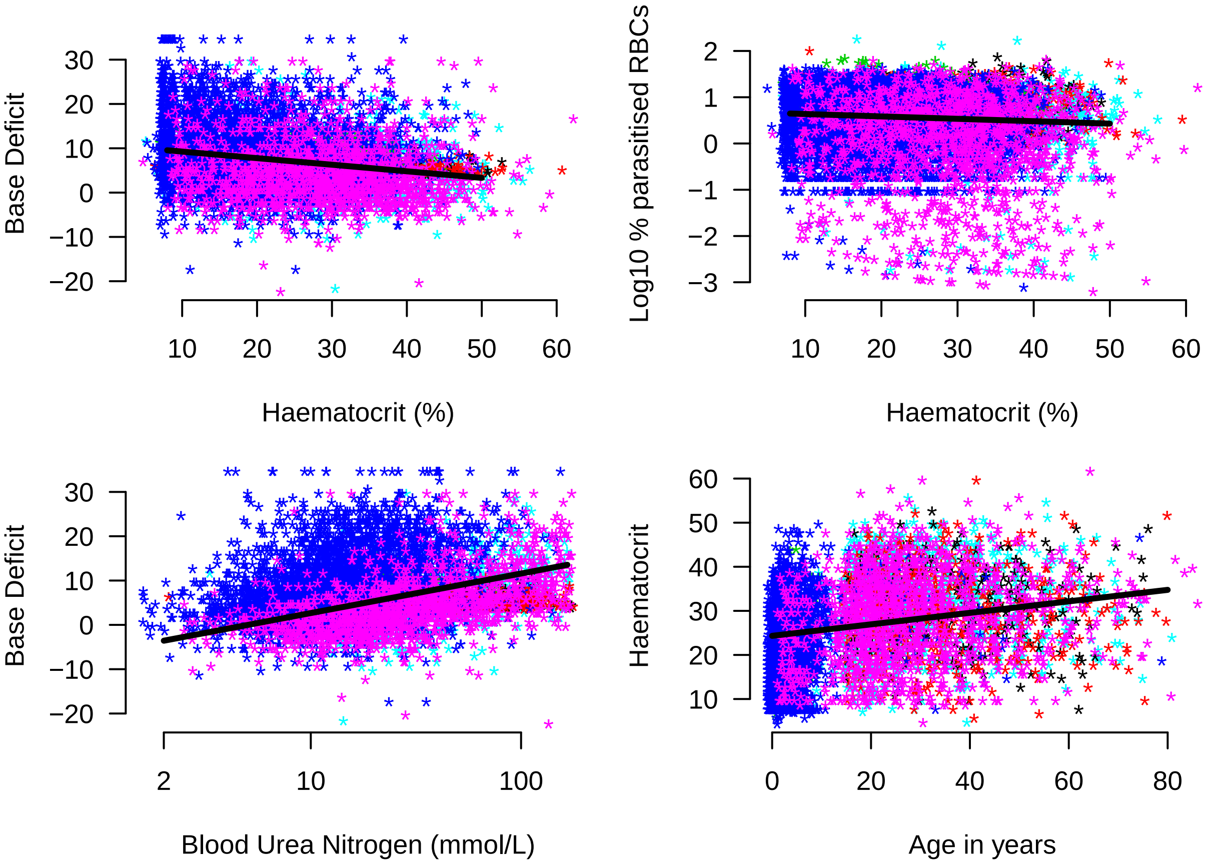
Let's look at the key predictive variables. We use a random effects term to model differences between studies.

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: BD ~ HCT + (1 | studyID)
##    Data: Complete_Leg_data
##
## REML criterion at convergence: 40270.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.2718 -0.6533 -0.1155  0.4845  4.4633
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  studyID  (Intercept)  1.366   1.169
##  Residual             43.095   6.565
## Number of obs: 6097, groups:  studyID, 6
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 10.736827   0.564028   19.04
## HCT         -0.148243   0.009565  -15.50
##
## Correlation of Fixed Effects:
##     (Intr)
## HCT -0.462
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: LPAR_pct ~ HCT + (1 | studyID)
##    Data: Complete_Leg_data
##
## REML criterion at convergence: 13843.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6726 -0.5624  0.1636  0.7317  2.5347
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  studyID  (Intercept) 0.07889  0.2809
##  Residual             0.56353  0.7507
## Number of obs: 6097, groups:  studyID, 6
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  0.688655   0.119797   5.749
## HCT         -0.005186   0.001097  -4.730
##
## Correlation of Fixed Effects:
##     (Intr)
## HCT -0.250
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: BD ~ log10(BUN) + (1 | studyID)
##    Data: Complete_Leg_data
##
## REML criterion at convergence: 39409.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.3679 -0.6217 -0.1011  0.5030  4.9640
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  studyID  (Intercept)  9.535   3.088
##  Residual             37.400   6.116
## Number of obs: 6097, groups:  studyID, 6
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  -6.2579     1.3229   -4.73
## log10(BUN)    8.9224     0.2581   34.57
##
## Correlation of Fixed Effects:
##            (Intr)
## log10(BUN) -0.283
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: HCT ~ AgeInYear + (1 | studyID)
##    Data: Complete_Leg_data
##
## REML criterion at convergence: 43682.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0759 -0.7518 -0.0554  0.6999  3.5504
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  studyID  (Intercept)  7.884   2.808
##  Residual             75.364   8.681
## Number of obs: 6097, groups:  studyID, 6
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 24.33535    1.19322   20.39
## AgeInYear    0.13029    0.01157   11.26
##
## Correlation of Fixed Effects:
##           (Intr)
## AgeInYear -0.220
```



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_GLM = glmer(outcome ~ HCT + LPAR_pct + AgeInYear + coma + convulsions +
                   poedema + log10(BUN) + BD + drug_AS + (1 | studyID),
               data = Complete_Leg_data, family=binomial)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : Model failed to converge with max|grad| = 0.00332676 (tol =
## 0.001, component 1)
```

```
summary(mod_full_GLM)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## outcome ~ HCT + LPAR_pct + AgeInYear + coma + convulsions + poedema +
##     log10(BUN) + BD + drug_AS + (1 | studyID)
##    Data: Complete_Leg_data
##
##      AIC      BIC   logLik deviance df.resid
##   3517.3   3591.2  -1747.7   3495.3     6086
##
## Scaled residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -4.6301 -0.3453 -0.1991 -0.1135 13.4583
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  studyID (Intercept) 0.01538  0.124
## Number of obs: 6097, groups:  studyID, 6
##
## Fixed effects:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.684523   0.290160 -23.037  < 2e-16 ***
## HCT           0.014509   0.005198   2.791 0.005255 **
## LPAR_pct      0.025881   0.060318   0.429 0.667867
## AgeInYear     0.016073   0.003694   4.351 1.36e-05 ***
## coma          1.367891   0.098267  13.920  < 2e-16 ***
## convulsions1  0.533741   0.112813   4.731 2.23e-06 ***
## poedema1      0.741742   0.370003   2.005 0.044996 *
## log10(BUN)    1.654044   0.160767  10.288  < 2e-16 ***
## BD            0.123580   0.007105  17.394  < 2e-16 ***
## drug_AS      -0.334685   0.089278  -3.749 0.000178 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) HCT    LPAR_p AgInYr coma   cnvls1 poedm1 l10(BU BD
## HCT         -0.510
## LPAR_pct    -0.027 -0.005
## AgeInYear    0.018 -0.143 -0.060
## coma        -0.149 -0.083  0.081 -0.093
## convulsins1 -0.122 -0.055  0.020  0.116 -0.252
## poedema1     0.009 -0.012  0.003 -0.101  0.025 -0.001
## log10(BUN)  -0.722  0.086 -0.082 -0.238 -0.045  0.116 -0.031
## BD          -0.193  0.235 -0.174  0.215 -0.004  0.047  0.018 -0.216
## drug_AS     -0.086 -0.017 -0.011 -0.077  0.007 -0.009 -0.021 -0.060 -0.020
## convergence code: 0
## Model failed to converge with max|grad| = 0.00332676 (tol = 0.001, component 1)
```

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

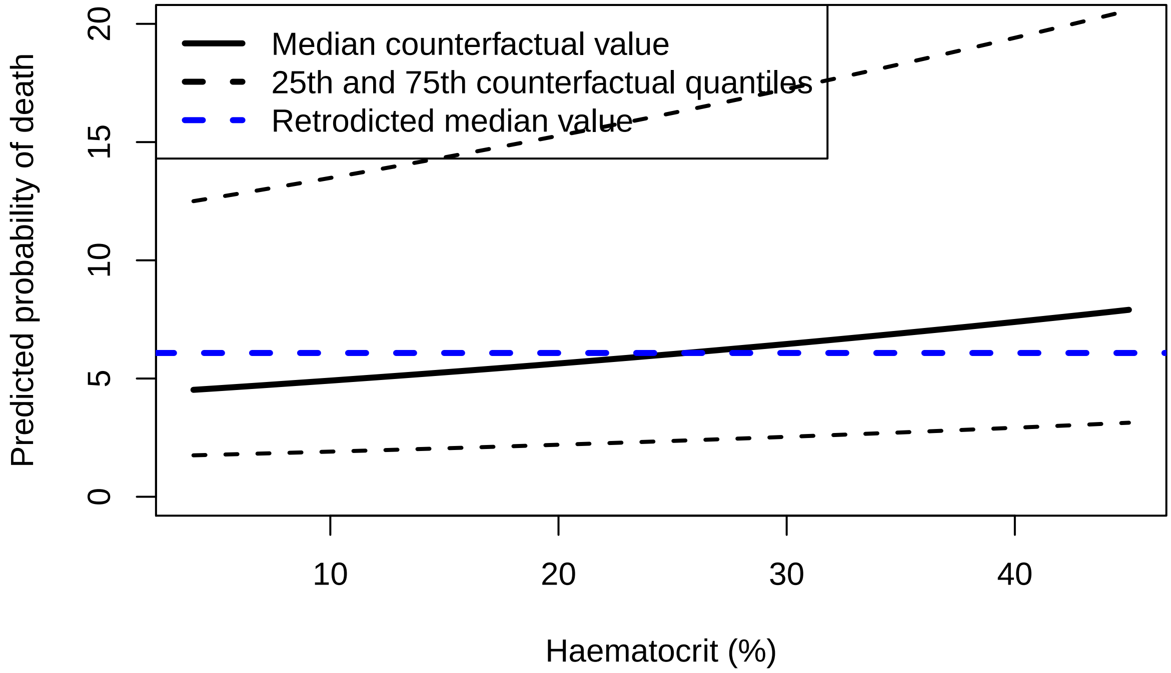
```
myquantiles = c(0.25,0.5,0.75) # this is 50% predictive interval

overall_median_mortality = median(100*predict(mod_full_GLM, 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_GLM, 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 + log10(BUN) + BD + drug_AS + s(studyID, bs='re'),
                 data=Complete_Leg_data, family=binomial)
summary(mod_full_GAM)
```

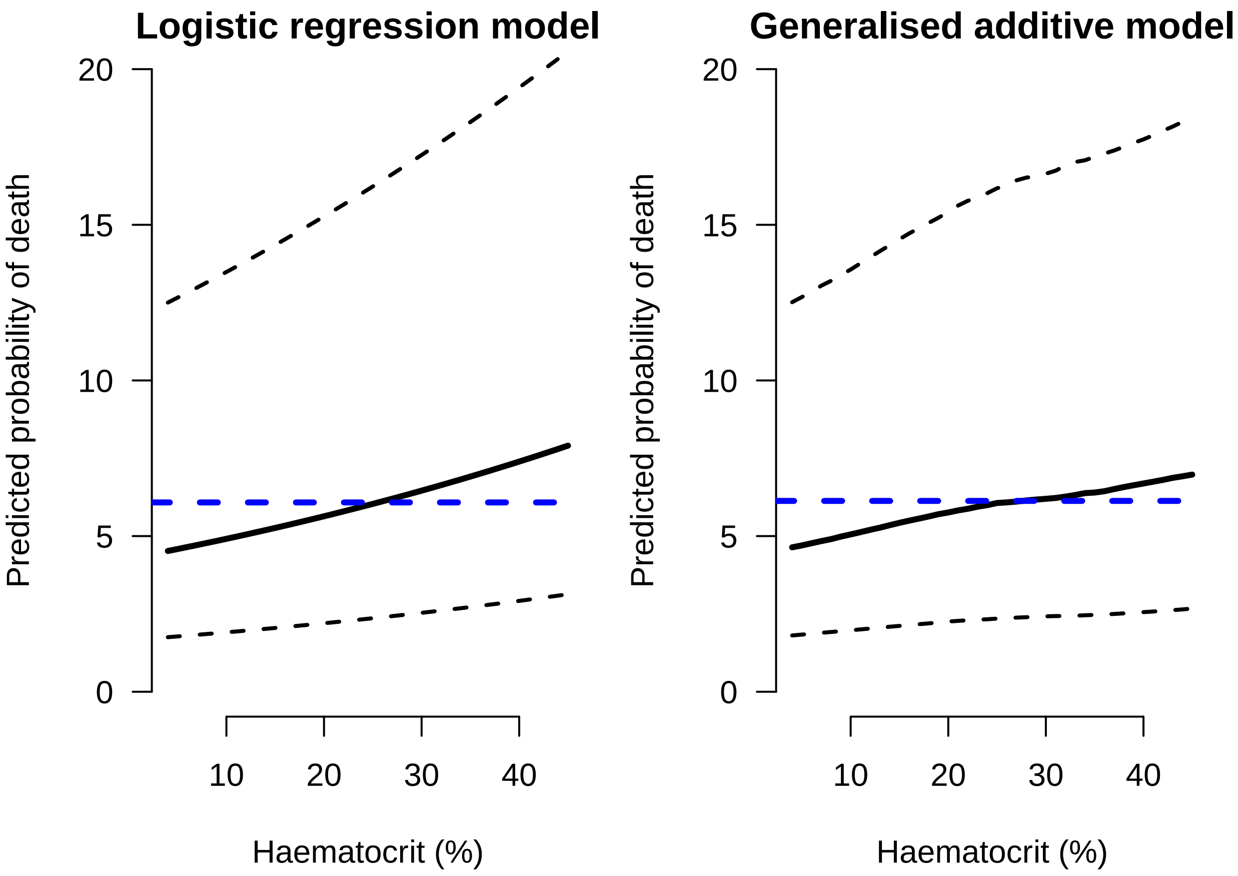
```
##
## Family: binomial
## Link function: logit
##
## Formula:
## outcome ~ s(HCT, AgeInYear) + LPAR_pct + coma + convulsions +
##     poedema + log10(BUN) + BD + drug_AS + s(studyID, bs = "re")
##
## Parametric coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.071683   0.260934 -23.269  < 2e-16 ***
## LPAR_pct      0.031316   0.059610   0.525 0.599343    
## coma          1.371608   0.097356  14.089  < 2e-16 ***
## convulsions1  0.553654   0.113722   4.868 1.12e-06 ***
## poedema1      0.750426   0.368134   2.038 0.041504 *  
## log10(BUN)    1.555950   0.164804   9.441  < 2e-16 ***
## BD            0.125944   0.007354  17.126  < 2e-16 ***
## drug_AS      -0.336923   0.089442  -3.767 0.000165 ***
## ---
## 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) 5.869  8.205 41.856 1.88e-06 ***
## s(studyID)       2.394  5.000  4.198    0.117    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## R-sq.(adj) =  0.254   Deviance explained = 27.2%
## UBRE = -0.42495  Scale est. = 1         n = 6097
```

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

```r
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:

```r
#
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')
```



Model comparison

Which model is better fit in terms of AIC

```r
print(AIC(mod_full_GAM, mod_full_GLM))
```

```
##                    df      AIC
## mod_full_GAM 16.26342 3506.094
## mod_full_GLM 11.00000 3517.329
```

And in terms of deviance

```r
print(list(deviance(mod_full_GLM), deviance(mod_full_GAM)))
```

```
## [[1]]
## [1] 3488.172
##
## [[2]]
## [1] 3473.568
```