Bias correction in GLMs

Bicko, Jonathan & Ben

2021 Jun 21 (Mon)

Introduction

We intend to investigate our prediction based on known truth and any bias potentially introduced by non-linear averaging, conditioning or random effect. We'll start with a simple case of a only fixed effect model and then consider a mixed effect model.

Simulation

We perform a simple simulation for a fixed effect model

$$
\begin{aligned}
\text{logit}(\text{status} = 1) &= \eta \\
\eta &= \beta_0 + \beta_{\text{A}}\text{Age} + \beta_{\text{W}}\text{Wealthindex} \\
\text{Age} &\sim \text{Normal}(0.2, 1) \\
\text{Wealthindex} &\sim \text{Normal}(0, 1) \\
\beta_0 &= 0.7 \\
\beta_{\text{A}} &= 0.3 \\
\beta_{\text{W}} &= 0.6
\end{aligned}
$$

```
N <- 1e4
beta0 <- 0.7
betaA <- 0.2
betaW <- 0.5

age_max <- 1
age_min <- 0.2
age <- runif(N, age_min, age_max)
# age <- rnorm(N, age_max, age_max)

wealthindex <- rnorm(N, 0, 1)

eta <- beta0 + betaA * age + betaW * wealthindex
sim_df <- (data.frame(age=age, wealthindex=wealthindex, eta=eta)
    %>% mutate(status = rbinom(N, 1, plogis(eta)))
    %>% select(-eta)
)
true_prop <- mean(sim_df$status)
print(true_prop)
```

```
## [1] 0.6916
```

```r
head(sim_df)
```

```
##         age wealthindex status
## 1 0.8452918   1.1198420      1
## 2 0.2563395  -0.6219684      0
## 3 0.4192913  -1.5949657      1
## 4 0.7882493  -1.2565989      1
## 5 0.3893671   1.7148530      1
## 6 0.8806260  -0.1938844      1
```

Simple logistic model

```r
simple_mod <- glm(status ~ age + wealthindex, data = sim_df, family="binomial")
```

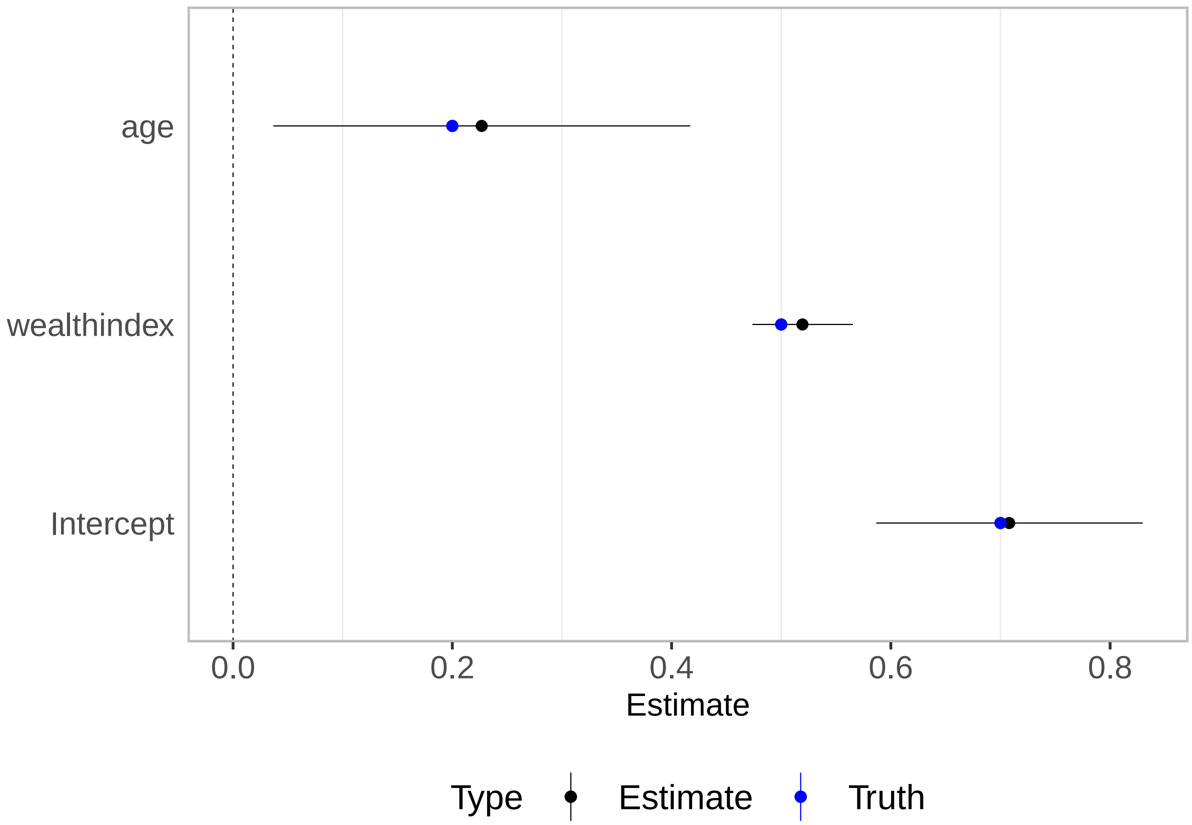
Coefficient plots

```r
## True beta
true_beta_df <- data.frame(term=c("Intercept", "age", "wealthindex")
    , estimate=c(beta0, betaA, betaW)
)

## Tidy coef estimates
coef_df <- (broom::tidy(simple_mod, conf.int=TRUE)
#   %>% dotwhisker::by_2sd(sim_df)
    %>% mutate(term = gsub("\\(|\\)", "", term))
)
print(coef_df)
```

```
## # A tibble: 3 x 7
##   term        estimate std.error statistic   p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 Intercept      0.708    0.0620     11.4  3.46e- 30   0.587      0.830
## 2 age            0.227    0.0970      2.34 1.94e-  2   0.0367     0.417
## 3 wealthindex    0.519    0.0234     22.2  5.09e-109   0.474      0.565
```

```r
simple_coef_plot <- (plotEsize(coef_df)
    + geom_point(data=true_beta_df, aes(x=term, y=estimate, colour="Truth"))
    + labs(colour="Type")
    + scale_colour_manual(values=c("black", "blue"))
)
print(simple_coef_plot)
```



Effect sizes on logit scale:

```
coef_df_logit <- (coef_df
    %>% select(term, estimate, conf.low, conf.high)
    %>% group_by(term)
    %>% summarise_all(plogis)
)
print(coef_df_logit)
```

```
## # A tibble: 3 x 4
##   term        estimate conf.low conf.high
##   <chr>          <dbl>    <dbl>     <dbl>
## 1 age            0.556    0.509     0.603
## 2 Intercept      0.670    0.643     0.696
## 3 wealthindex    0.627    0.616     0.638
```

Variable predictions

We consider both **varpred** and population averaging approach; and then introduce bias correction to **varpred** predictions.

```
popavefun <- function(mod, focal, non.focal, level=0.95, steps=100, modelname="quantiles", ...) {
    mf <- model.frame(mod)
    mm <- (mf
        %>% select_at(c(focal, non.focal))
    )
    quant <- seq(0, 1, length.out=steps)
    mm <- sapply(c(focal, non.focal), function(x)quantile(mm[,x], quant), simplify = FALSE)
    mm <- do.call("expand.grid", mm)
```

```
    vc <- vcov(mod)
    mm2 <- model.matrix(formula(mod)[c(1,3)], mm)
    linpred <- as.vector(mm2 %*% coef(mod))
    pse_var <- sqrt(rowSums(mm2 * t(tcrossprod(data.matrix(vc), mm2))))
    z.val <- qnorm(1 - (1 - level)/2)
    pred_df <- (mm
        %>% select_at(focal)
        %>% mutate(fit = linpred
            , lwr = plogis(fit - z.val*pse_var)
            , upr = plogis(fit + z.val*pse_var)
            , fit = plogis(fit)
        )
        %>% group_by_at(focal)
        %>% summarise_at(c("fit", "lwr", "upr"), mean)
        %>% mutate(model = modelname, se=NA)
    )
    return(pred_df)
}

binfun <- function(mod, focal, non.focal, bins=50, ...) {
    mf <- model.frame(mod)
    mm <- (mf
        %>% select_at(c(focal, non.focal))
    )
    check_df <- (mf
        %>% arrange_at(focal)
        %>% mutate(bin=ceiling(row_number()*bins/nrow(.)))
        %>% group_by(bin)
        %>% summarise_all(mean)
        %>% mutate(model="binned")
    )
    return(check_df)
}
```

- Age

```
## varpred way
simple_vareff_age <- varpred(simple_mod, "age", isolate=FALSE, modelname="varpred")

## Pop. average
simple_vareff_age_pop <- varpred(simple_mod, "age", isolate=FALSE, pop.ave="quantile", modelname="quant
binned_df <- binfun(simple_mod, "age", "wealthindex")

vareff_age <- simple_vareff_age
vareff_age$preds <- do.call("rbind", list(vareff_age$preds, simple_vareff_age_pop$preds))
age_plot <- (plot(vareff_age)
    + labs(y="Prob. of improved \n service", colour="Model")
    + geom_hline(yintercept=true_prop, lty=2, colour="grey")
    + geom_point(data=binned_df, aes(x=age, y=status, color="binned"))
    + scale_colour_manual(values=c("black", "blue", "red", "green"))
    + theme(legend.position="bottom")
)
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```

- Wealth index

```
# Wealth index
## varpred
simple_vareff_wealthindex <- varpred(simple_mod, "wealthindex", isolate=FALSE, modelname="varpred")

## Pop. average
simple_vareff_wealthindex_pop <- varpred(simple_mod, "wealthindex", isolate=FALSE, pop.ave="quantile", m
binned_df <- binfun(simple_mod, "wealthindex", "age")

vareff_wealthindex <- simple_vareff_wealthindex
vareff_wealthindex$preds <- do.call("rbind", list(vareff_wealthindex$preds, simple_vareff_wealthindex_po
wealthindex_plot <- (plot(vareff_wealthindex)
    + labs(y="", colour="Model")
    + geom_hline(yintercept=true_prop, lty=2, colour="grey")
    + geom_point(data=binned_df, aes(x=wealthindex, y=status, color="binned"))
    + scale_colour_manual(values=c("black", "blue", "red"))
    + theme(legend.position="bottom")
)
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```

```
ggarrange(age_plot, wealthindex_plot, common.legend=TRUE)
```

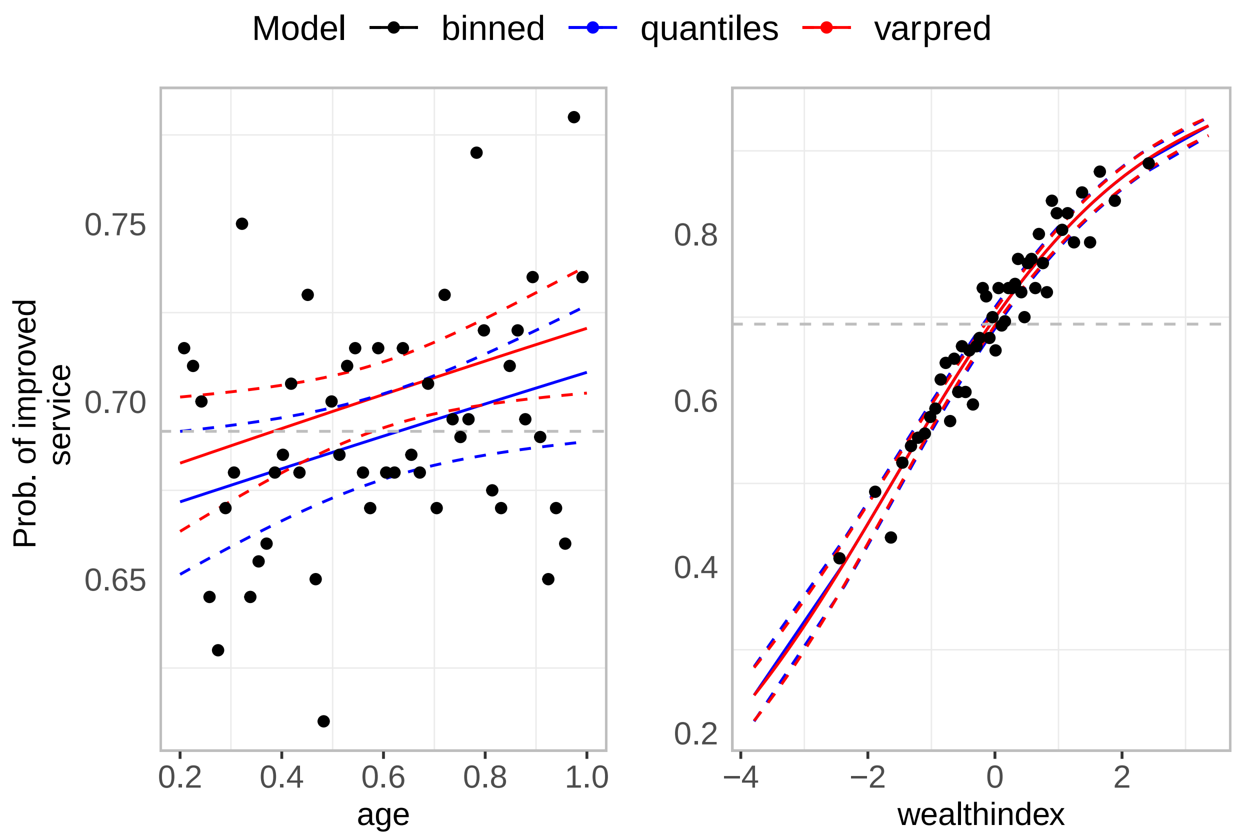


Figure 1: A comparison of population averaged and varpred-based predictions. In the case of biased predictions (age), population averaging gives better estimates. On the other hand, for wealthindex, the estimates varpred and population estimates are somehow similar.

The observed population average is 0.6916 while the estimated population averages (similar estimates with `varpred`) are:

- age: pop. average 0.690167; varpred 0.7018437
- wealthindex: pop. average 0.690167; varpred 0.6404765

Random effect model

```
# Simulation parameters
nHH <- 1000 # Number of HH (primary units) per year

nyrs <- 50  # Number of years to simulate
yrs <- 2000 + c(1:nyrs) # Years to simulate
N <- nyrs * nHH

## HH random effect sd
hhSD <- 0.5

# Generate dataset template
temp_df <- (data.frame(hhid = rep(c(1:nHH), each = nyrs)
        , years = rep(yrs, nHH)
        , age = runif(n=N, age_min, age_max)
        , wealthindex = rnorm(n = N, 0, 1)
    )
)

# Simulate HH-level random effects (residual error)
hhRE <- rnorm(nHH, hhSD)
temp_df$hhRE <- hhRE[temp_df$hhid]

sim_df <- (temp_df
    %>% mutate(eta = beta0 + betaA * age + betaW * wealthindex + hhRE
        , status = rbinom(N, 1, plogis(eta))
    )
    %>% select(-eta)
)
true_prop_reff <- mean(sim_df$status)
print(true_prop)
```

```
## [1] 0.6916
```

```
print(head(sim_df, 10))
```

```
##    hhid years       age wealthindex      hhRE status
## 1     1  2001 0.9018686 -1.37824486 -1.961837      0
## 2     1  2002 0.3506389 -0.01121371 -1.961837      0
## 3     1  2003 0.4490254  0.85993585 -1.961837      0
## 4     1  2004 0.9666503  1.09017090 -1.961837      0
## 5     1  2005 0.5382138 -2.02888505 -1.961837      0
## 6     1  2006 0.2568336 -0.35194828 -1.961837      0
## 7     1  2007 0.9850446  0.84014258 -1.961837      1
## 8     1  2008 0.2927210 -0.10084596 -1.961837      1
## 9     1  2009 0.8826939  0.10647195 -1.961837      1
## 10    1  2010 0.2453546  0.57340856 -1.961837      0
```

Fit model

```
reff_mod <- glmmTMB(status ~ age + wealthindex + (1|hhid)
    , data = sim_df
    , family = binomial(link = "logit")
)
```

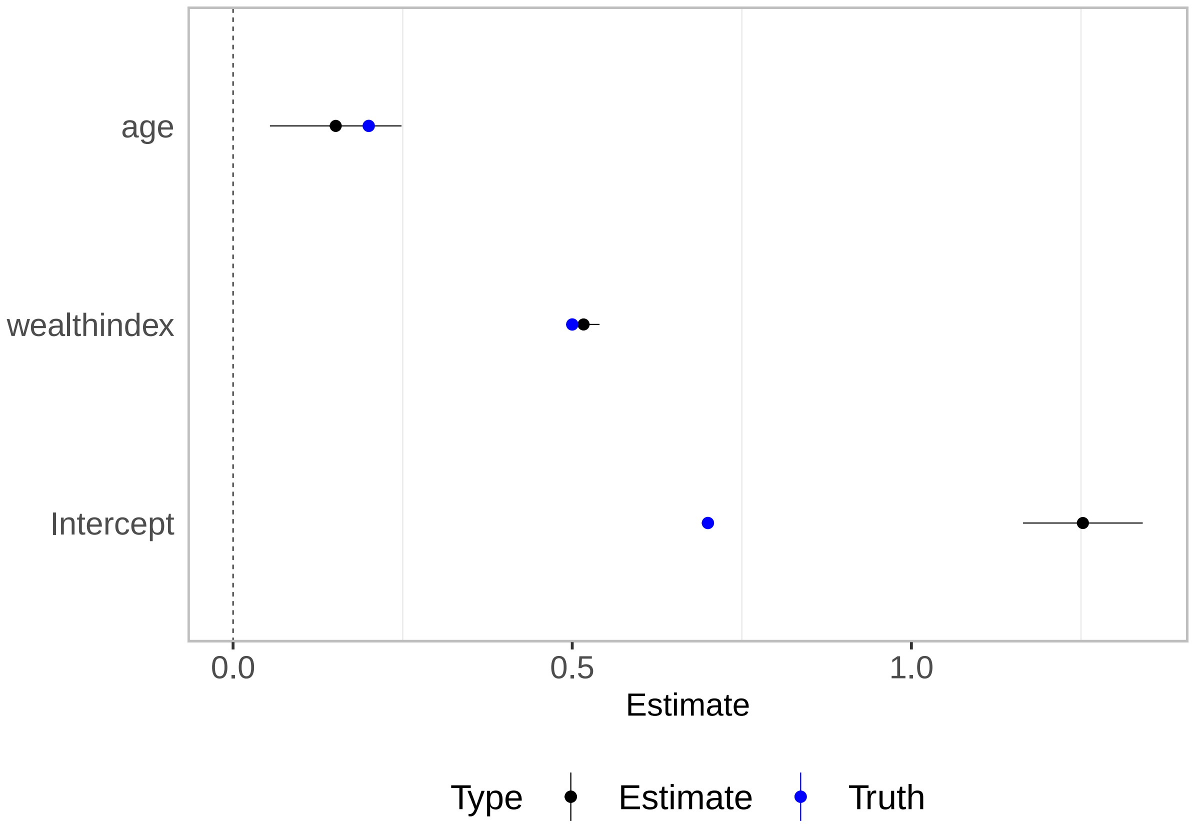
```
## Tidy coef estimates
reff_coef_df <- (broom.mixed::tidy(reff_mod, conf.int=TRUE)
    %>% mutate(term = gsub("\\(|\\)", "", term))
    %>% filter(effect=="fixed")
)
```

```
## Registered S3 method overwritten by 'broom.mixed':
##   method      from
##   tidy.gamlss broom
```

```
print(reff_coef_df)
```

```
## # A tibble: 3 x 10
##   effect component group term    estimate std.error statistic   p.value conf.low
##   <chr>  <chr>     <chr> <chr>      <dbl>     <dbl>     <dbl>     <dbl>    <dbl>
## 1 fixed  cond      <NA>  Interc~    1.25     0.0450     27.8  2.30e-170   1.16
## 2 fixed  cond      <NA>  age        0.151    0.0495      3.06 2.23e-  3   0.0543
## 3 fixed  cond      <NA>  wealth~    0.517    0.0120     43.0  0.          0.493
## # ... with 1 more variable: conf.high <dbl>
```

```
reff_coef_plot <- (plotEsize(reff_coef_df)
    + geom_point(data=true_beta_df, aes(x=term, y=estimate, colour="Truth"))
    + labs(colour="Type")
    + scale_colour_manual(values=c("black", "blue"))
)
print(reff_coef_plot)
```



Variable effect plots

- Age

```
## varpred way
reff_vareff_age <- varpred(reff_mod, "age", isolate=FALSE, modelname="varpred")

## Pop. average
reff_vareff_age_pop <- varpred(reff_mod, "age", isolate=FALSE, pop.ave="quantile", include.re=TRUE, mode
binned_df <- binfun(reff_mod, "age", "wealthindex")

vareff_age <- reff_vareff_age
vareff_age$preds <- do.call("rbind", list(vareff_age$preds, reff_vareff_age_pop$preds))
age_plot <- (plot(vareff_age)
    + labs(y="Prob. of improved \n service", colour="Model")
    + geom_hline(yintercept=true_prop_reff, lty=2, colour="grey")
    + geom_point(data=binned_df, aes(x=age, y=status, color="binned"))
    + scale_colour_manual(values=c("black", "blue", "red"))
    + theme(legend.position="bottom")
)
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```

- Wealth index

```
# Wealth index
## varpred
reff_vareff_wealthindex <- varpred(reff_mod, "wealthindex", isolate=FALSE, modelname="varpred")
binned_df <- binfun(reff_mod, "wealthindex", "age")
```

```
## Pop. average
reff_vareff_wealthindex_pop <- varpred(reff_mod, "wealthindex", isolate=FALSE, pop.ave="quantile", incl

vareff_wealthindex <- reff_vareff_wealthindex
vareff_wealthindex$preds <- do.call("rbind", list(vareff_wealthindex$preds, reff_vareff_wealthindex_pop$
wealthindex_plot <- (plot(vareff_wealthindex)
    + labs(y="", colour="Model")
    + geom_hline(yintercept=true_prop_reff, lty=2, colour="grey")
    + geom_point(data=binned_df, aes(x=wealthindex, y=status, color="binned"))
    + scale_colour_manual(values=c("black", "blue", "red"))
    + theme(legend.position="bottom")
)
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```

```
ggarrange(age_plot, wealthindex_plot, common.legend=TRUE)
```

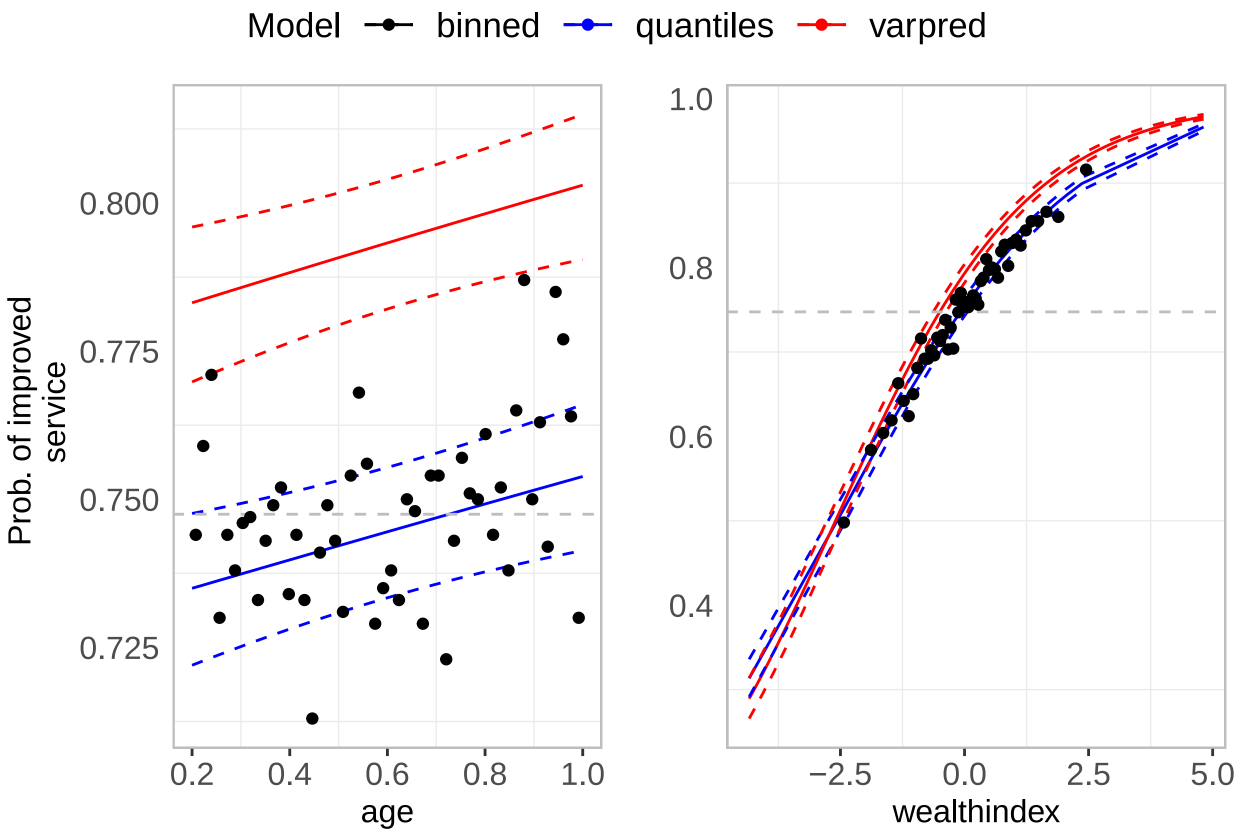


Figure 2: A comparison of quantile and varpred based predictions. In both cases, varpred seems to over-approximate the predictions.