Bias correction in GLMs

Bicko, Jonathan & Ben

2021 Jun 21 (Mon)

Introduction

Prediction of the variation in the response variable depends on whether the relationship between the response variable and the predictors is linear or nonlinear. For example, when response, Y, changes nonlinearly with the predictor variable, X, averaged response variable with respect to the predictor, $E(Y(X))$, does not necessarily equal the response at the mean of the predictor, $Y(E(X)$. This can be understood in relation to Jensen's inequality which states that, for a nonlinear function, $Y(X)$, then $E(Y(X)) > Y(E(X))$, if $Y(X)$ is positive second derivative; and $E(Y(X)) < Y(E(X))$ if $Y(X)$ is negative second derivative. Most packages for predicting responses make distribution assumptions about the predictors, (for example, conditioning at the mean value of the predictor), leading to potential biasness if the particular predictor is not well represented, or as a result of nonlinear averaging. We consider the following approaches for bias correction:

- Population averaging
  - Whole population
  - Quantiles
- Distributional conditioning
- Second-order correction

We implement and apply these methods in the context of both simple generalized linear and mixed effect models, using simulated data sets.

Simple fixed effect model

$$\begin{aligned}
\text{logit}(\text{status} = 1) &= \eta \\
\eta &= \beta_0 + \beta_{\text{A}}\text{Age} + \beta_{\text{W}}\text{Wealthindex} \\
\text{Age} &\sim \text{Normal}(0.2, 1) \\
\text{Wealthindex} &\sim \text{Normal}(0, 1) \\
\beta_0 &= 0.7 \\
\beta_{\text{A}} &= 0.2 \\
\beta_{\text{W}} &= 0.5
\end{aligned}$$

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

age_max <- 1
age_min <- 0.2
```

```r
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)
```

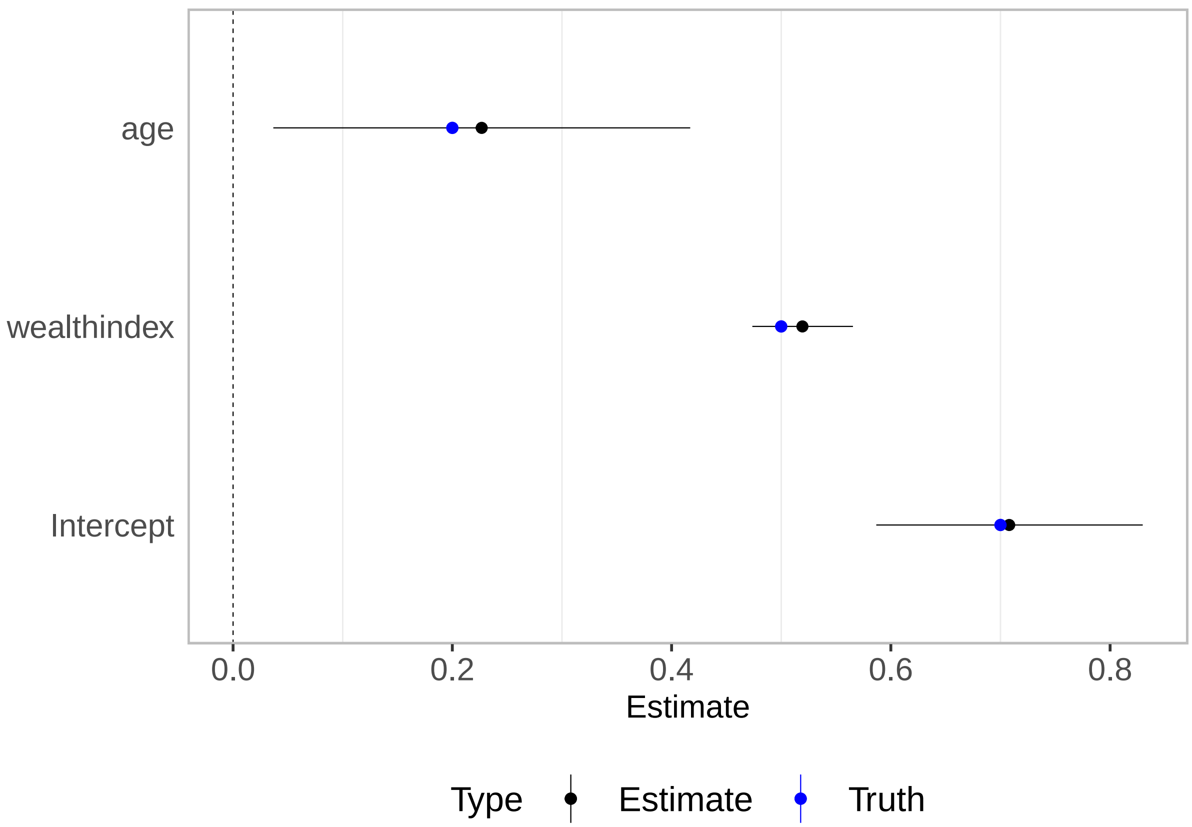

Figure 1: Coefficient plot for simple logistic model.

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.

- Age

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

```r
## Pop. average
### Quantiles
simple_vareff_age_quant <- varpred(simple_mod, "age", isolate=FALSE, pop.ave="quantile", modelname="quan
### Population
simple_vareff_age_pop <- varpred(simple_mod, "age", isolate=FALSE, pop.ave="population", modelname="Popu
binned_df <- binfun(simple_mod, "age", "wealthindex", bins=15)

vareff_age <- simple_vareff_age
vareff_age$preds <- do.call("rbind", list(vareff_age$preds, simple_vareff_age_quant$preds, simple_vareff
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("grey", "blue", "red", "black"))
    + theme(legend.position="bottom")
)
```

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

- Wealth index

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

## Pop. average
### Quantiles
simple_vareff_wealthindex_quant <- varpred(simple_mod, "wealthindex", isolate=FALSE, pop.ave="quantile"
### Population
simple_vareff_wealthindex_pop <- varpred(simple_mod, "wealthindex", isolate=FALSE, pop.ave="population"
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_qu
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("grey", "blue", "red", "black"))
    + theme(legend.position="bottom")
)
```

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

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

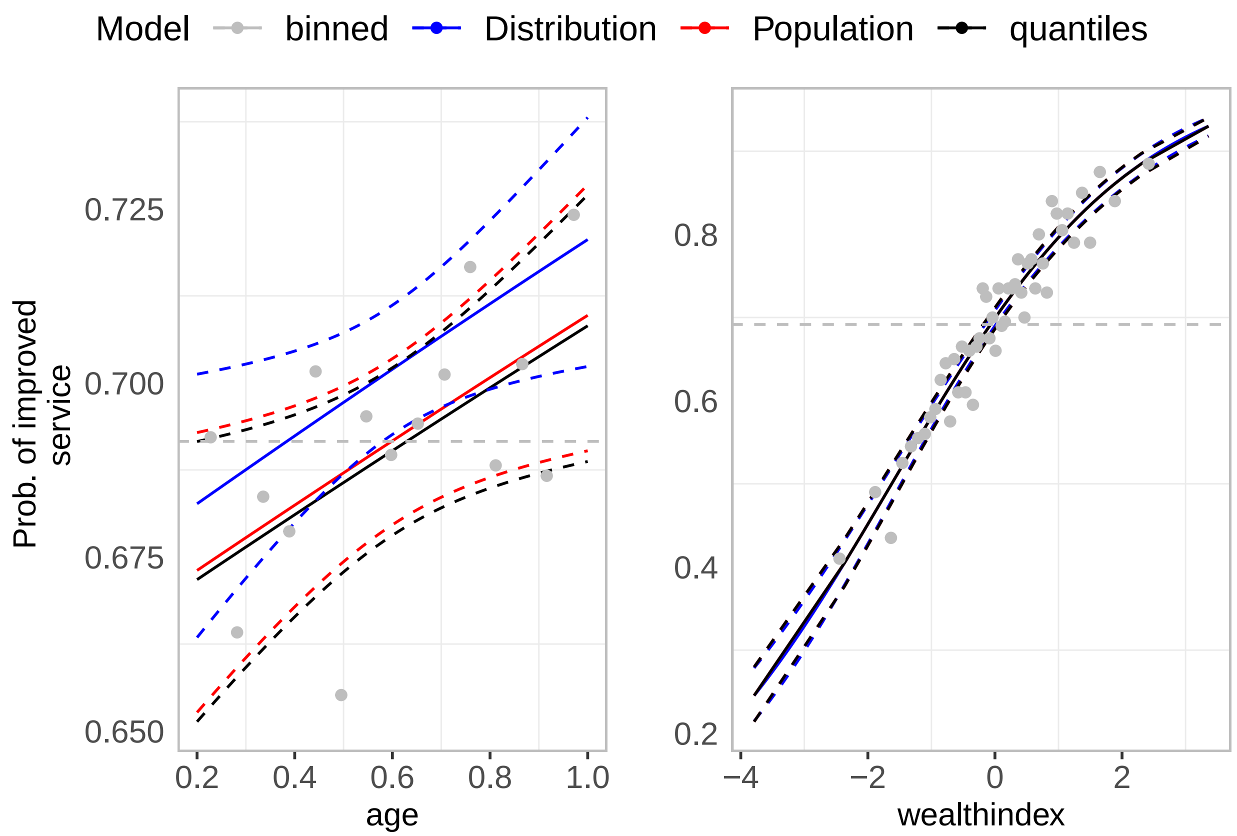

Figure 2: A comparison of various bias-correction approaches – Distribution approach is based on averaging of non-focal predictors; Population approach involves using the observed values of non-focal predictors together with the quantiles of the focal predictors; while quantiles approach involves sampling (or picking quantiles of) the observed values of both focal and non-focal predictors. Except for the distribution approach, population and quantiles approaches give very close estimates.

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

- age: quantiles 0.690167; population 0.6915805; distribution 0.7018437
- wealthindex: quantiles 0.690167; population 0.6901678; distribution 0.6404765

Random effect model

```
# Simulation parameters
nHH <- 100  # 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)
```

```r
        , 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
```

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

```
##    hhid years       age wealthindex     hhRE status
## 1     1  2001 0.9018686 -0.99696269 0.514887      1
## 2     1  2002 0.3506389  0.68378134 0.514887      1
## 3     1  2003 0.4490254 -0.18218552 0.514887      1
## 4     1  2004 0.9666503  0.01820703 0.514887      1
## 5     1  2005 0.5382138 -1.48422812 0.514887      1
## 6     1  2006 0.2568336  1.06422463 0.514887      0
## 7     1  2007 0.9850446  0.08233927 0.514887      1
## 8     1  2008 0.2927210 -1.02046500 0.514887      0
## 9     1  2009 0.8826939  2.11619121 0.514887      1
## 10    1  2010 0.2453546 -0.31856268 0.514887      1
```

Fit model

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

```r
## 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
```

```r
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>  Interce~    1.30      0.142      9.19  3.90e-20   1.02
## 2 fixed cond      <NA>  age         0.139     0.159      0.872 3.83e- 1  -0.173
## 3 fixed cond      <NA>  wealthi~    0.548     0.0386    14.2   9.77e-46   0.472
## # ... with 1 more variable: conf.high <dbl>
```

```r
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)
```

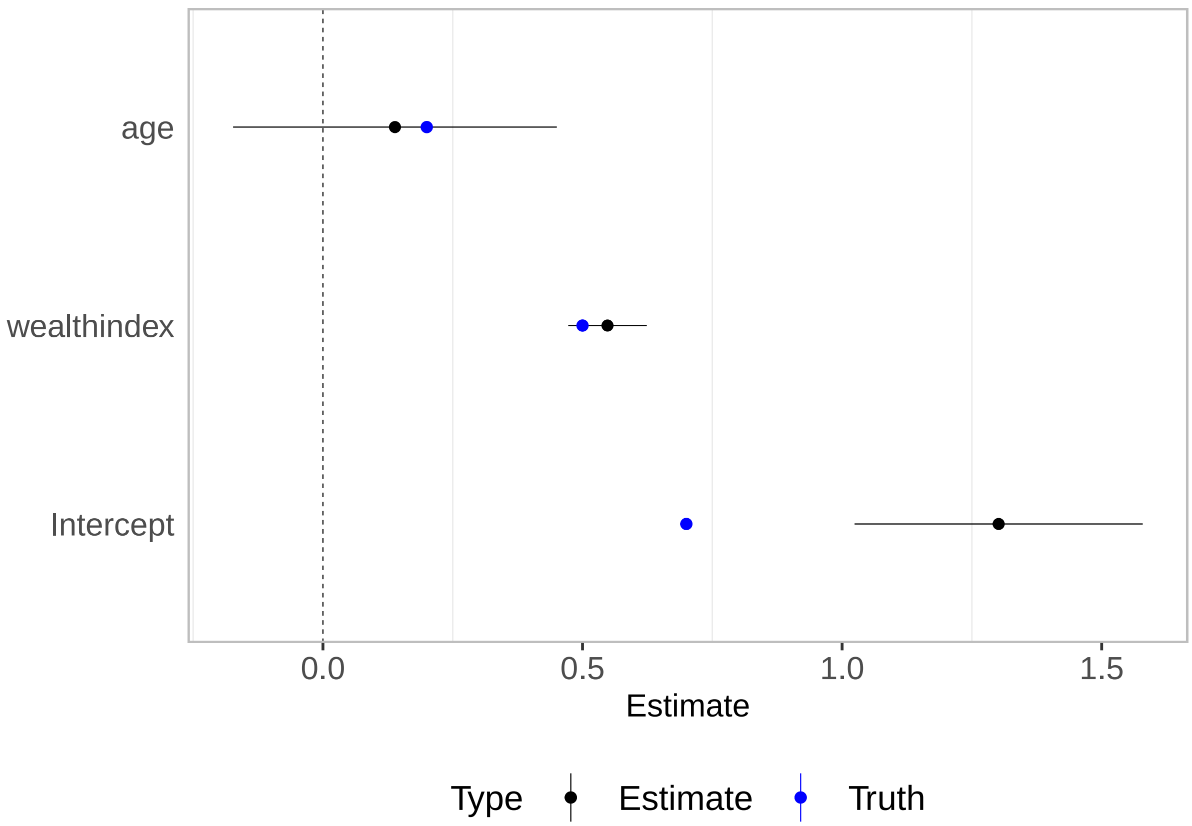

Figure 3: Mixed model coefficient estimates.

Variable effect plots

- Age

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

## Pop. average
### Quantiles
reff_vareff_age_quant <- varpred(reff_mod, "age", isolate=FALSE, pop.ave="quantile", include.re=TRUE, mo
### Population
reff_vareff_age_pop <- varpred(reff_mod, "age", isolate=FALSE, pop.ave="population", include.re=TRUE, mo
binned_df <- binfun(reff_mod, "age", "wealthindex", bins=15)
```

```
vareff_age <- reff_vareff_age
vareff_age$preds <- do.call("rbind", list(vareff_age$preds, reff_vareff_age_quant$preds, reff_vareff_ag
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("grey", "blue", "red", "black"))
    + 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="Distribution")
binned_df <- binfun(reff_mod, "wealthindex", "age")

## Pop. average
### Quantiles
reff_vareff_wealthindex_quant <- varpred(reff_mod, "wealthindex", isolate=FALSE, pop.ave="quantile", in
### Population
reff_vareff_wealthindex_pop <- varpred(reff_mod, "wealthindex", isolate=FALSE, pop.ave="population", in

vareff_wealthindex <- reff_vareff_wealthindex
vareff_wealthindex$preds <- do.call("rbind", list(vareff_wealthindex$preds, reff_vareff_wealthindex_qua
vareff_wealthindex$preds <- do.call("rbind", list(vareff_wealthindex$preds, reff_vareff_wealthindex_qua
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("grey", "blue", "red", "black"))
    + 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)
```

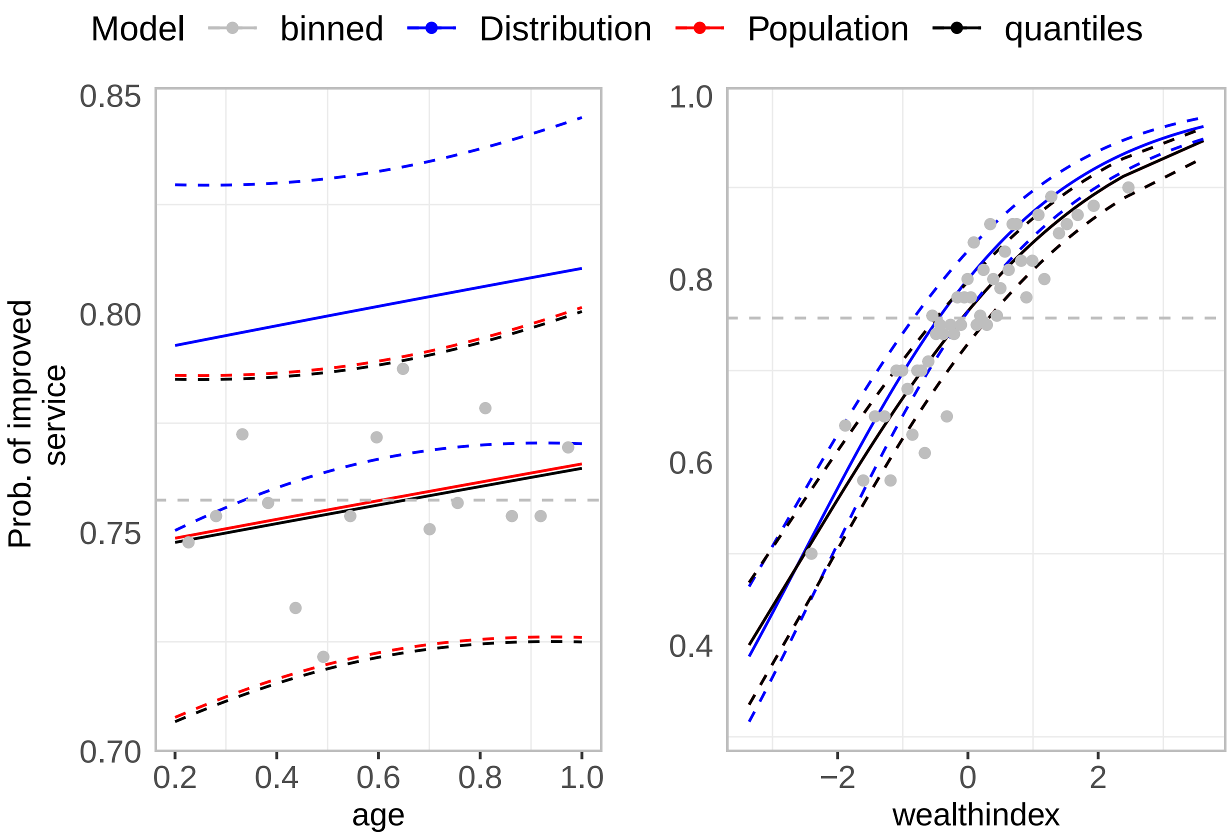

Figure 4: Distribution based approach over-estimates the predictions in age and wealth index.