PIJ project update - AUC values, and simpler models

ute hahn

16.4.2021

Purposes

- get calibration adjustment
- Calculate AUC (was missing) for SE model
- compare with some simpler models

Data preparation

```
suppressPackageStartupMessages(library("tidyverse"))
```

Eriks fitted model

```
load("../cache/model_reduced_lean.RData")
varlist_SE <- attr(model_reduced_lean$terms, "term.labels")
levels_SE <- model_reduced_lean$xlevels
```

Danish data, with indices

```
datadir <- "../../data/"

dkdata_x <- haven::read_sas(paste0(datadir, "working_on_final.sas7bdat")) %>%
  rename(P_Age = age_index) %>%
  mutate(
    outcome = as.logical(outcome5_90d),
    P_BMI = cut(OBMI,
                breaks = c(0, 25, 30, 35, 50),
                labels = levels_SE$P_BMI,
                right = FALSE),
    P_Sex = factor(KOEN, levels = 2:1, labels = levels_SE$P_Sex),
    P_DiaGrp = factor(diagnosis_grp,
                      labels = levels_SE$P_DiaGrp),
    P_ASA = factor(OASA_SCORE, labels = levels_SE$P_ASA)
  ) %>%
  select(c("outcome", all_of(varlist_SE), "CCI_comorb", "Elix_comorb", "Rx_index")) %>%
  mutate(across(starts_with("c_"), as.logical)) %>%
  mutate(CCI = as.factor(CCI_comorb), Elix = as.factor(Elix_comorb), Rx = as.factor(Rx_index))
```

Steyerberg recalibration

As Erik suggested, the Swedish model should be recalibrated when validated with Danish data. Here is the resulting shift, in case it is needed later

```r
Z <- predict(model_reduced_lean, dkdata_x, type = "response")

fit2 <- glm(dkdata_x$outcome ~ 1, offset = Z)
fit2$coefficients
```

```
## (Intercept)
## -0.00349242
```

AUC of Eriks original Swedish model

We did not have information on the actual AUC last time. I wrapped the whole thing into a function for later use on other models. I chose the asymptotic confidence interval here to save computing time - anyway, it does not affect the actual estimate.

```r
auc_out <- function(model, data = dkdata_x){
  pred <- predict(model, data, type = "response")
  result <- as.vector(pROC::ci.auc(data$outcome, pred, levels = c('FALSE', "TRUE"),
                                   direction = "<", method = "delong"))
  names(result) <- c("ci.lo", "auc.est", "ci.hi")
  result[c(2,1,3)]
}

auc_out(model_reduced_lean)
```

```
##   auc.est     ci.lo     ci.hi
## 0.6644978 0.6383293 0.6906663
```

Compare several models

Below are AUC results for some more models. Although the Danish data differ in some way from the Swedish, in particular in the number of childhood hip disease diagnoses, the original fitted values from Sweden perform suprisingly well, and refitting the same model to Denmark only gives little improvement.

I have also included the models from Figure 3, that only take ASA or comorbidity index into account. The AUC values for the comorbidity index models are roughly 10% lower than Eriks results. If run on Danish data with the standard components age and sex, the AUC improves and compares to the Swedish results in Figure 3. I did handle them as numeric variables, though.

Finally, I included just for curiosity also a model that has no medical covariables, and another one that does not include any diagnoses apart from ASA score. Interestingly, the "laymen" model with BMI performs better than those that do not account for BMI, even though there is no medical knowledge involved. But this is only based on a rough AUC value.

```r
models <- list(
  "SE model refitted to DK" = model_reduced_lean$formula,
  "ASA only" = "P_ASA",
  "ASA + Age + Sex" = "P_ASA + P_Age + P_Sex",
  "Elixhauser only" = "Elix_comorb",
  "Elixhauser + Age + Sex" = "Elix_comorb + P_Age + P_Sex",
  "Charlson only" = "CCI_comorb",
  "Charlson + Age + Sex" = "CCI_comorb + P_Age + P_Sex",
  "Rx Risk only" = "Rx_index",
  "Rx Risk + Age + Sex" = "Rx_index + P_Age + P_Sex",
  "laymen: BMI + Age + Sex" = "P_BMI + P_Sex + P_Age",
  "No diags: ASA + BMI + Age + Sex" = "P_ASA + P_BMI + P_Sex + P_Age"
```

```
) %>%
 map(~ ifelse(grepl("~", .x), .x, paste("outcome ~", .x))) %>%
 map(~ glm(as.formula(.x), family = binomial(), data = dkdata_x))

models <- c("SE model" = list(model_reduced_lean),
            "SE recalibrated" = list(fit2),
            models)

modelnames <- names(models)

t(sapply(models[-2], auc_out))
```

```
##                                   auc.est     ci.lo     ci.hi
## SE model                        0.6644978 0.6383293 0.6906663
## SE model refitted to DK         0.6718784 0.6458812 0.6978755
## ASA only                        0.5827459 0.5593311 0.6061608
## ASA + Age + Sex                 0.6050408 0.5795086 0.6305730
## Elixhauser only                 0.5199970 0.4986380 0.5413560
## Elixhauser + Age + Sex          0.5730051 0.5457002 0.6003100
## Charlson only                   0.5279087 0.5083930 0.5474245
## Charlson + Age + Sex            0.5731801 0.5460877 0.6002724
## Rx Risk only                    0.5791562 0.5504993 0.6078130
## Rx Risk + Age + Sex             0.6023291 0.5768198 0.6278383
## laymen: BMI + Age + Sex         0.6398137 0.6126092 0.6670181
## No diags: ASA + BMI + Age + Sex 0.6464293 0.6199900 0.6728687
```

Refitted model

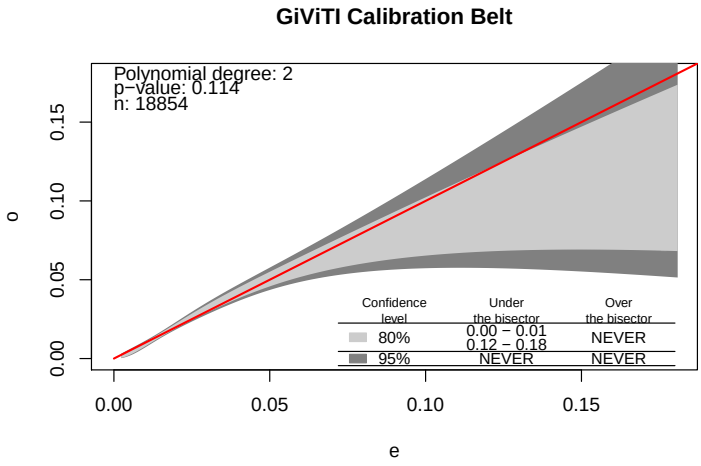
As we discussed last time, the "Table 1"s in both countries differ, and the fitted models have a big discrepancy with respect to the coefficient of sequelae on childhood hip disease. But as Erik pointed out in his follow up mail, we should not put too much interpretation into the coefficients and consider the model more as a kind of prediction device.

If one were concerned about different coefficients in other countries, one could refit the model. It does not change much overall in the prediction, though.

Below is the calibration curve for the model, when it is refitted to the Danish data.

```
obspred <-
  tibble(
    obs  = as.integer(dkdata_x$outcome),
    pred_refit = predict(models[[3]], dkdata_x, type = "response"),
    pred_recalib = predict(models[[2]], dkdata_x, type = "response")
  )
```

```
 calibration <-
 givitiR::givitiCalibrationBelt(
   obspred$obs,
   obspred$pred_refit,
   devel = "internal"
 )

plot(calibration, xlim = c(0, 0.18), ylim = c(0, 0.18))
```

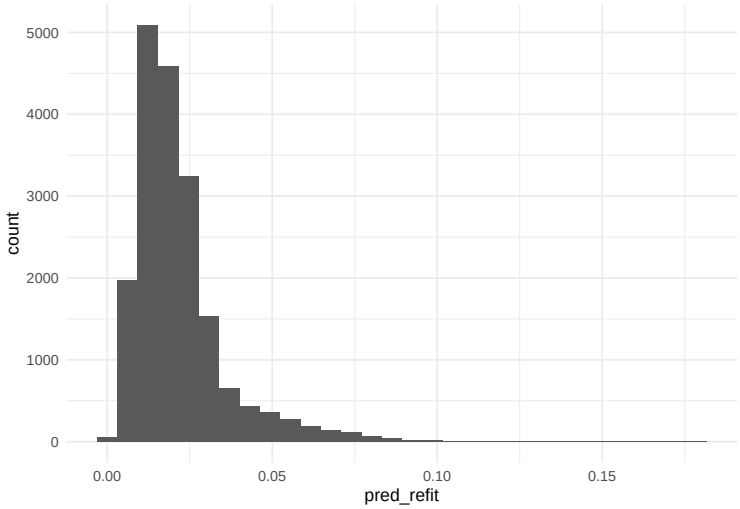



```
## $m
## [1] 2
##
## $p.value
## [1] 0.1142464
```

There are only few very high predicted values, so we could ignore the right part in the calibration plot

```
ggplot(data = obspred, aes(x = pred_refit)) +
  geom_histogram() +
  theme_minimal()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
map_int(c(0.05, 0.1, 0.15, 0.2), ~  sum(obspred$pred_refit > .x))
```

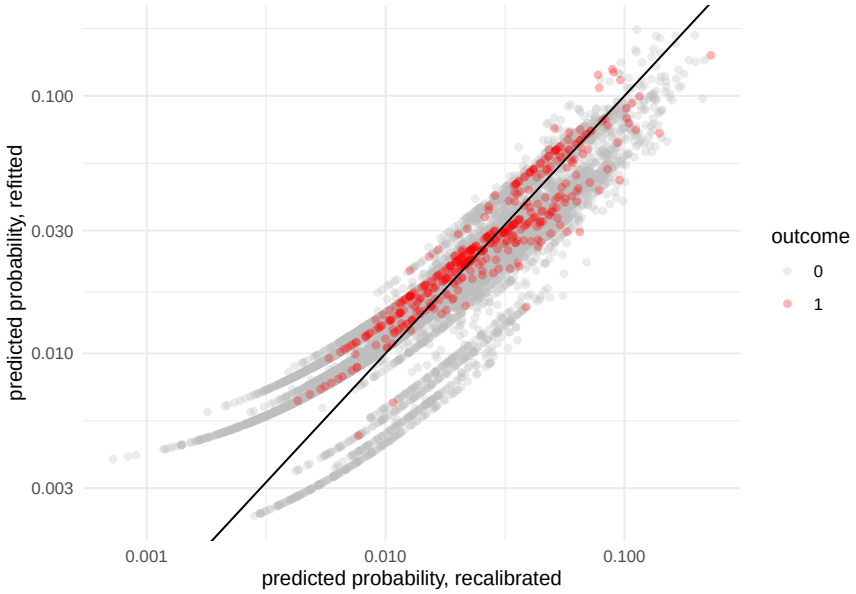
```
## [1] 1049   53    8    0
```

There is some difference between refitted and merely recalibrated predictions. The refitted model seems to have a tendency to report more high probabilities, in the cases where the recalibrated model gave a low probability. On the other hand, the tendency is reversed for high probabilities. Thus, the refitted model seems more moderate - if this is not a bug in my program, then it is a bit surprising, because the refitted

model should be closer to the data and thus more prone to overfitting (?)

The figure below is on log-log scale.

```
ggplot(data = obspred %>% arrange(obs),
       aes(x = pred_recalib, y=pred_refit,
                          colour = factor(obs)))+
  geom_point(alpha = 0.3) +
  theme_minimal()+
  scale_color_manual(values = c("grey", "red")) +
  labs(x = "predicted probability, recalibrated", y = "predicted probability, refitted",
       colour = "outcome") +
  scale_x_continuous(trans = "log10") +
  scale_y_continuous(trans = "log10") +
  geom_abline(slope = 1, intercept = 0)
```




```
with(obspred, mean(pred_refit > pred_recalib))
```

```
## [1] 0.6854248
```

```
with(obspred %>% filter(pred_recalib<0.005), mean(pred_refit > pred_recalib))
```

```
## [1] 0.9219949
```

```
with(obspred %>% filter(pred_recalib>0.1), mean(pred_refit > pred_recalib))
```

```
## [1] 0.1031746
```

Appendix: Predicted probabilities within different strata of the population

Density of predicted probabilities by stratum and outcome.

Red vertical line: estimated proportion of outcomes in stratum,

Black vertical line: average of the predicted values.

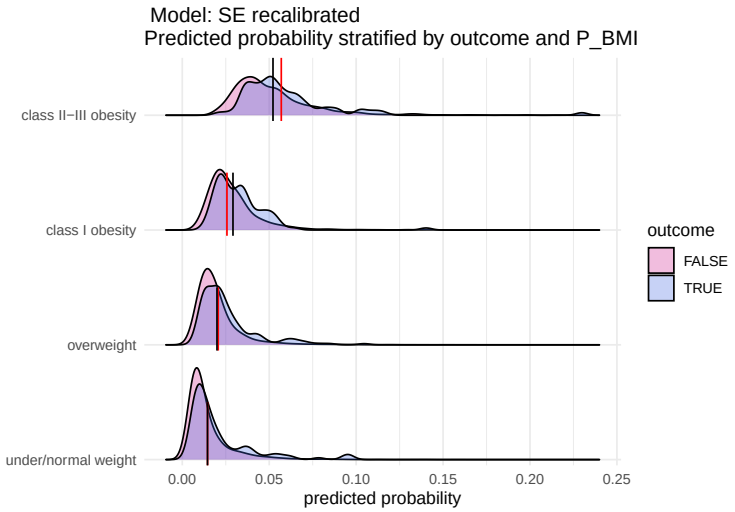
For the recalibrated Swedish model, we can detect a slight discrepancy between black and red line, other than for the reference group. This is due to slightly different estimated coefficients in Sweden.

The graphs are somewhat misleading in that the density curves represent different number of people. Therefore occasionally the proportion of people with high predicted probability looks too big, when it refers to a small subpopulation.

```
library(ggridges)

plotstratified <- function(stratum, modelno = 2){
  usedat <- tibble(
      outcome = dkdata_x$outcome,
      pred = predict(models[[modelno]], dkdata_x, type = "response"),
      stratum = as.factor(dkdata_x[[stratum]])
    )
  stratlevel <- levels(usedat$stratum)

  obsprob <- tibble(
    trueprop = tapply(usedat$outcome, usedat$stratum, mean),
    meanpred = tapply(usedat$pred, usedat$stratum, mean),
    stratum = seq_along(stratlevel))

  ggplot(usedat, aes(y=stratum)) +
    ggridges::geom_density_ridges(aes(x = pred,
                                      fill = paste(stratum,factor(outcome))),
                                  alpha = .3, scale = .8)+
    geom_segment(data = obsprob,
                 mapping = aes(x = trueprop, xend = trueprop,
                               y = as.numeric(stratum)-.05, yend = as.numeric(stratum)+.5),
                 inherit.aes = FALSE,
                 color = "red") +
    geom_segment(data = obsprob,
                 mapping = aes(x = meanpred, xend = meanpred,
                               y = as.numeric(stratum)-.05, yend = as.numeric(stratum)+.5),
                 inherit.aes = FALSE,
                 color = "black") +
    scale_fill_cyclical(values=c(" violetred", "royalblue"),
                        name = "outcome", guide="legend", labels = c("FALSE", "TRUE")) +
    scale_y_discrete(expand = c(0, 0)) +
    theme_minimal() + theme(axis.title.y = element_blank()) +
    xlab('predicted probability' ) +
    ggtitle(paste(' Model:',modelnames[modelno],'\nPredicted probability stratified by outcome and', st
}
```

```
plotstratified("P_BMI")
```

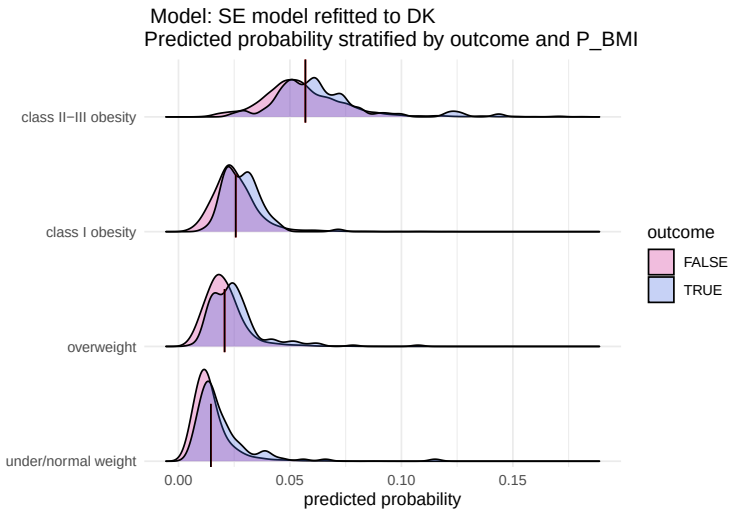
```
## Picking joint bandwidth of 0.00337
```



compare with refitted model: average of predicted probabilities coincide with observed proportion of outcomes

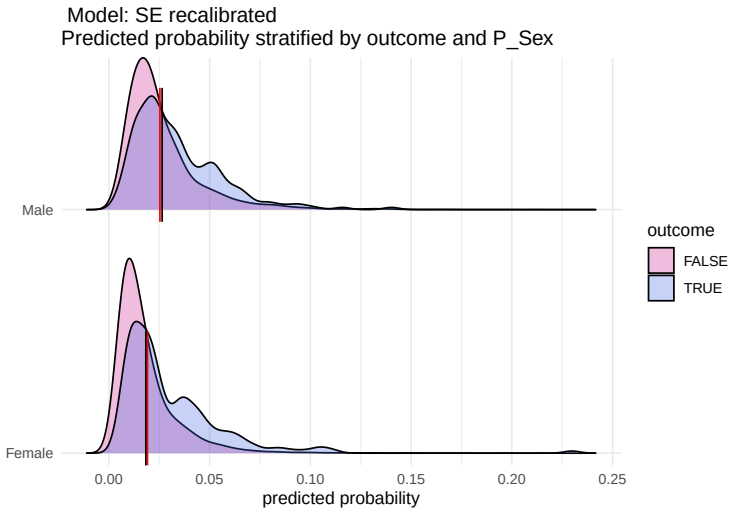
```
plotstratified("P_BMI", 3)
```

```
## Picking joint bandwidth of 0.00265
```



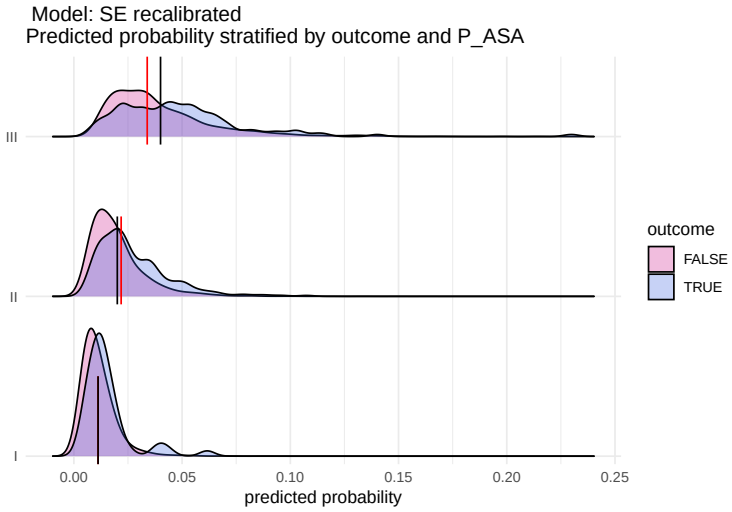

```
plotstratified("P_Sex")
```

```
## Picking joint bandwidth of 0.00392
```



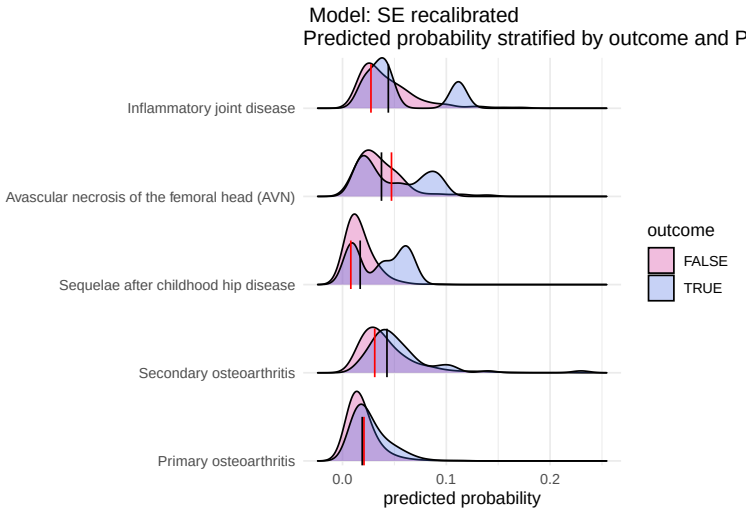

```
plotstratified("P_ASA")
```

```
## Picking joint bandwidth of 0.00349
```



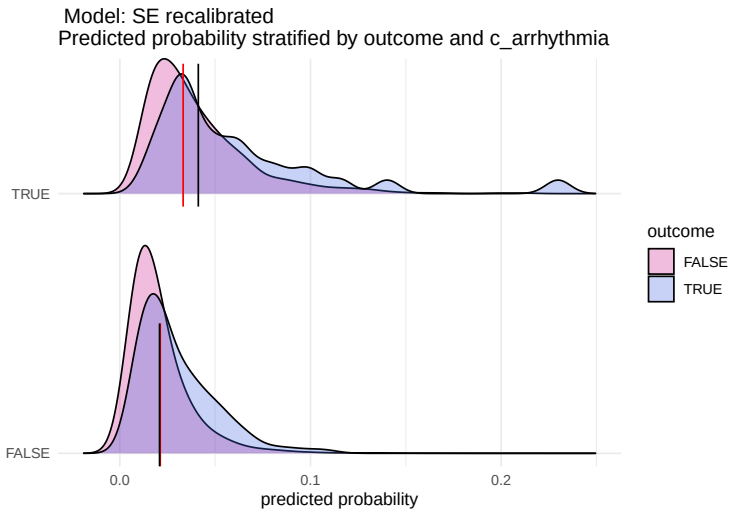

```
plotstratified("P_DiaGrp")
```

```
## Picking joint bandwidth of 0.00822
```



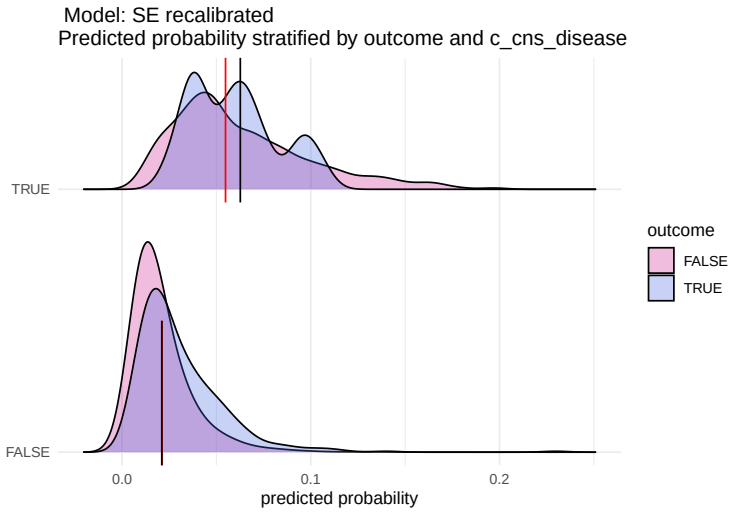

```
plotstratified("c_arrhythmia")
```

```
## Picking joint bandwidth of 0.00655
```



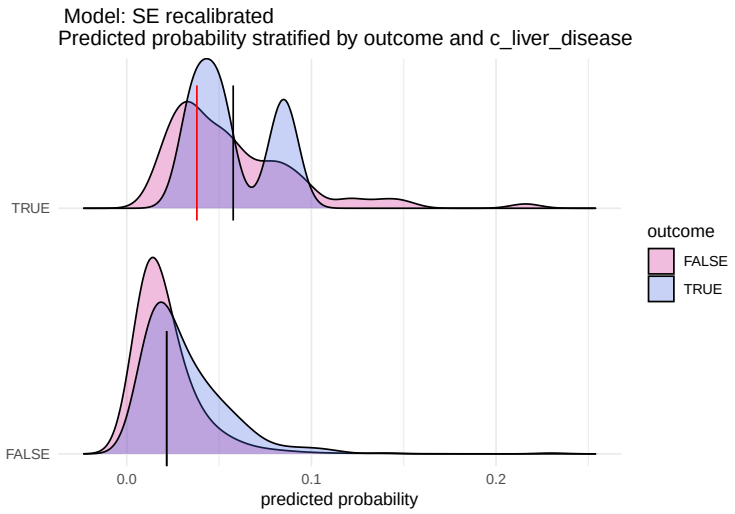

```
plotstratified("c_cns_disease")
```

```
## Picking joint bandwidth of 0.007
```



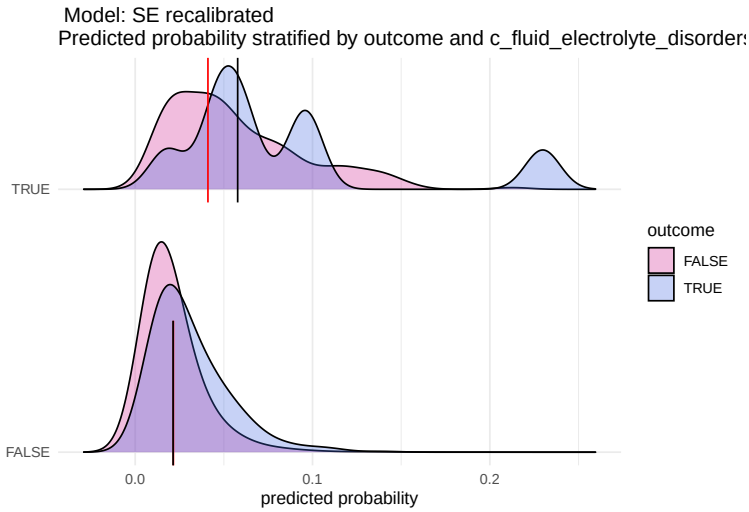

```
plotstratified("c_liver_disease")
```

```
## Picking joint bandwidth of 0.00802
```




```
plotstratified("c_fluid_electrolyte_disorders")
```

```
## Picking joint bandwidth of 0.00992
```




```
plotstratified("c_lung_airways_disease")
```

```
## Picking joint bandwidth of 0.00644
```