check_code_2.R

apontej

2021-02-04

```
# Make the simulations of the estimations depending on the number of categories
# 20200204 by JJAV
##############################################################################3

library(plyr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.5     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
```

```
library(rjags)
```

```
## Loading required package: coda
```

```
## Linked to JAGS 4.3.0
```

```
## Loaded modules: basemod,bugs
```

```
library(afdx)
library(doMC)
```

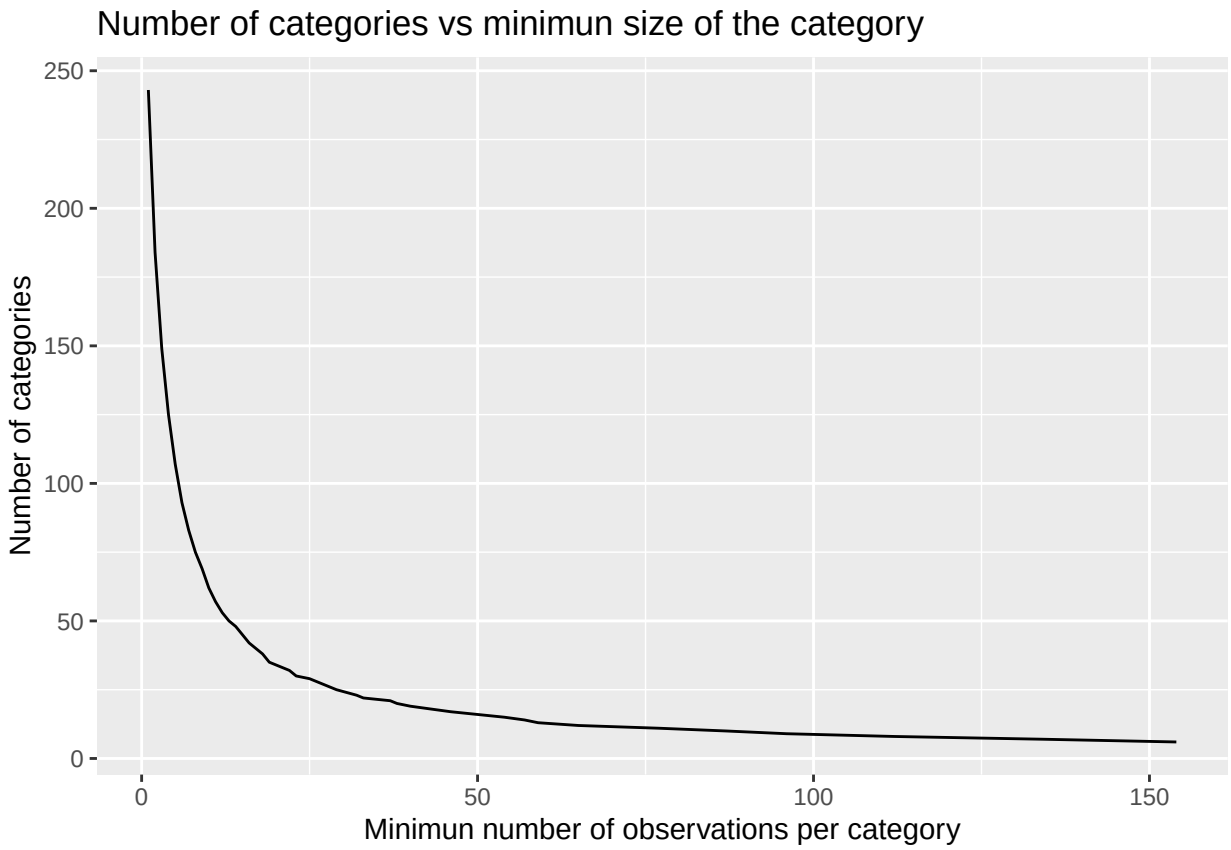
```
## Loading required package: foreach
```

```
##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when

## Loading required package: iterators

## Loading required package: parallel
```

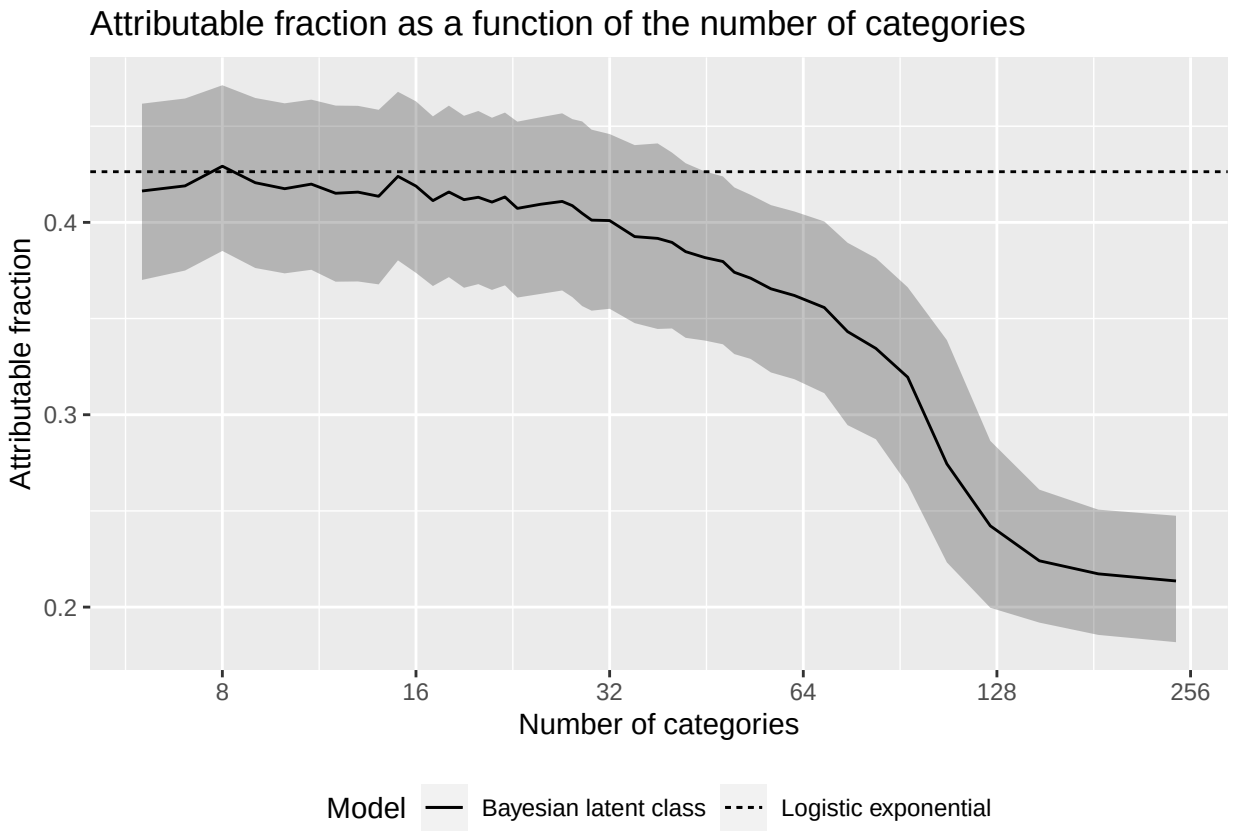
Estimate the AF for the malaria_df1 dataset using different categories

```
estimate_af <- function(nmin) {
  cat("[",nmin,"] " )

  # Make the data with cutoff points, having a nmin obs per categorie
  cutpoints <- make_n_cutoffs(malaria_df1$fever, malaria_df1$density, nmin)

  data <-
    malaria_df1 %>%
    mutate(k = cut(density,c(cutpoints,Inf),
                   include.lowest =T,
                   labels = cutpoints)) %>%
    group_by(k,fever) %>%
    tally() %>%
    mutate(category = ifelse(fever ==1,"fever","no_fever")) %>%
    select(-fever) %>%
    pivot_wider(
      names_from = "category",
      values_from = "n",
      values_fill = list(n = 0))


  # define the model
  jags_af = rjags::jags.model(
    textConnection(get_latent_model()),
    data=list(n=data$fever, m =data$no_fever),
    inits = list(.RNG.name = "base::Wichmann-Hill"),
    n.adapt=1000)

  # simulate the posterior
  jagssamples_af =
    coda.samples(jags_af,
    variable.names = c("lambda","sens","spec","ppv","npv"),
    n.iter = 10000,
    n.burnin=2000,
    n.thinning = 5)

  stats_af <- summary(jagssamples_af)

  # Return the analysis of the posterior
  data.frame(cbind(stats_af[[1]], stats_af[[2]])) %>%
```

```
    mutate(varname = row.names(.)) %>%
    mutate(ncuts = length(cutpoints)) %>%
    mutate(cutoff = c(NA, rep(cutpoints,4)))
}

# Select when there is a change in the number of categories
chgcat <- ldply(1:200, function(x){data.frame(nmin = x, ncats = length(make_n_cutoffs(malaria_df1$fever
  group_by(ncats) %>%
  filter(nmin == min(nmin)) %>%
  ungroup()

# Make the loop if the simulation does not exists
if (file.exists("sim_af_cats.rds")) {
   sim_af_cats <- readRDS("sim_af_cats.rds")
} else {
  registerDoMC(cores=12)
  sim_af_cats <- ddply(chgcat, .(nmin), function(x){estimate_af(x$nmin)},  .parallel = TRUE)
  saveRDS(sim_af_cats, file = "data-raw/sim_af_cats.rds")
}

# remove the 0 category
sim_af_cats <- sim_af_cats %>% filter(is.na(cutoff) | cutoff > 0 )

#logistic_exponential_model
fit <- logitexp(malaria_df1$fever, malaria_df1$density)
cutoffs <-  make_cutoffs(malaria_df1$fever, malaria_df1$density)
dxp <- senspec(fit, cutoffs[-1]) %>%
  data.frame(.) %>%
  pivot_longer(-cutoff, names_to = "Indicator",values_to = "Value")

# Plot the minimum number of observation per category vs the number of observations
sim_af_cats %>%
  select(nmin, ncuts) %>%
  unique()%>%
  ggplot(
    aes(x = nmin, y = ncuts)
  ) +
  geom_line()+
  scale_x_continuous("Minimun number of observations per category") +
  scale_y_continuous("Number of categories") +
  ggtitle("Number of categories vs minimun size of the category")
```



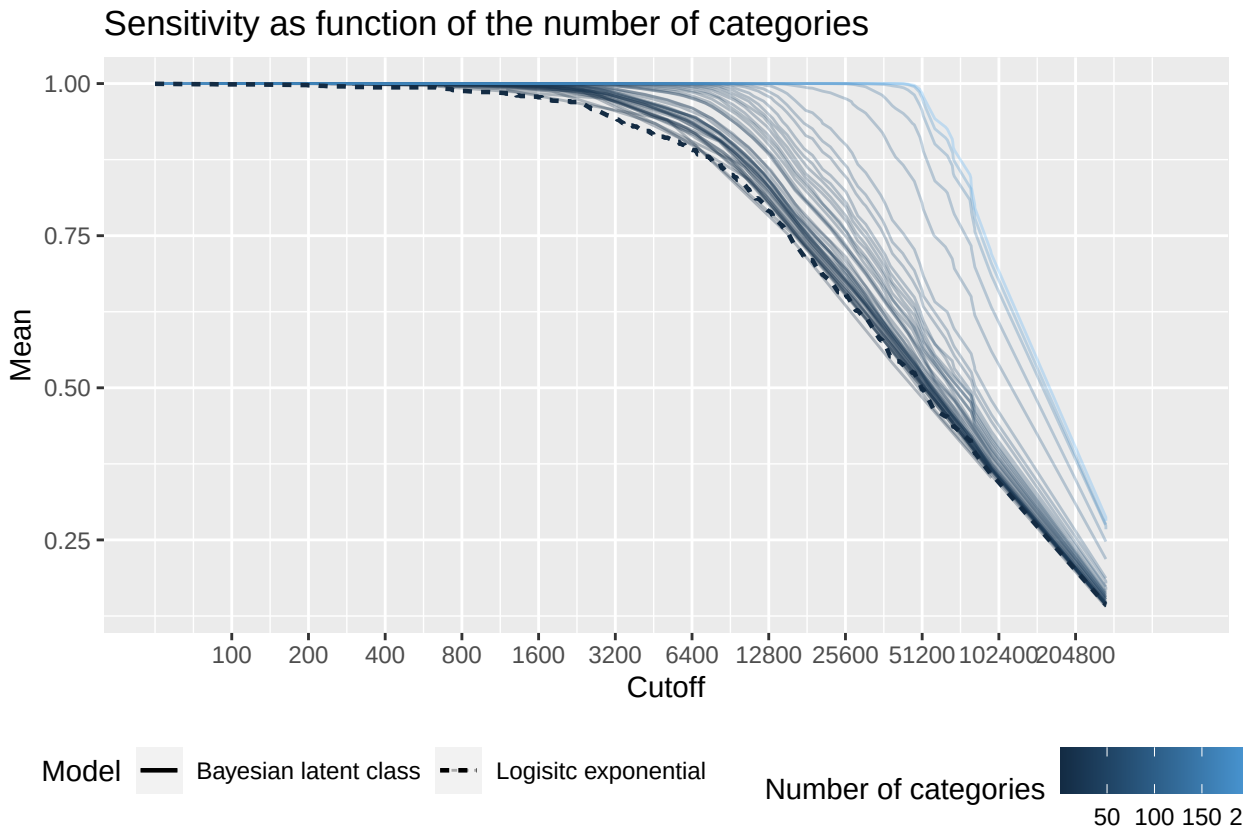

```
# Plot of the attributable fraction as a function of the number of categories
sim_af_cats %>%
  filter(varname == "lambda") %>%
  ggplot(
    aes(x = ncuts, y = Mean, ymin = X2.5., ymax = X97.5.)) +
  geom_ribbon(color = NA, alpha = 0.3) +
  geom_line(aes(linetype = "Bayesian latent class"))+
  geom_hline(aes(yintercept = fit$af ,linetype = "Logistic exponential")) +
  scale_x_log10("Number of categories",breaks = c(4,8,16,32,64,128,256)) +
  scale_y_continuous("Attributable fraction") +
  scale_linetype("Model") +
  ggtitle("Attributable fraction as a function of the number of categories") +
  theme(legend.position = "bottom")
```




```
# Plot of the sensitivites
sim_af_cats %>%
  filter(grepl("sens", varname)) %>%
  ggplot(
    aes(x = cutoff, y = Mean, group=ncuts, color = ncuts)) +
  geom_line(aes(linetype = "Bayesian latent class"), alpha = 0.3)+
  geom_line(data = dxp %>% filter(Indicator == "sensitivity") %>% mutate(ncuts = 1),
            aes(x = cutoff,
                y = Value,
                linetype = "Logisitc exponential",
                color = 1), size = 0.8 )+
  scale_x_log10("Cutoff", breaks=c(100,200,400,800,1600,3200,6400,12800,25600,51200,102400,204800), lim
  scale_linetype("Model") +
  scale_colour_continuous("Number of categories") +
  ggtitle("Sensitivity as function of the number of categories") +
  theme(legend.position = "bottom")
```

```
## Warning: Removed 74 row(s) containing missing values (geom_path).
```

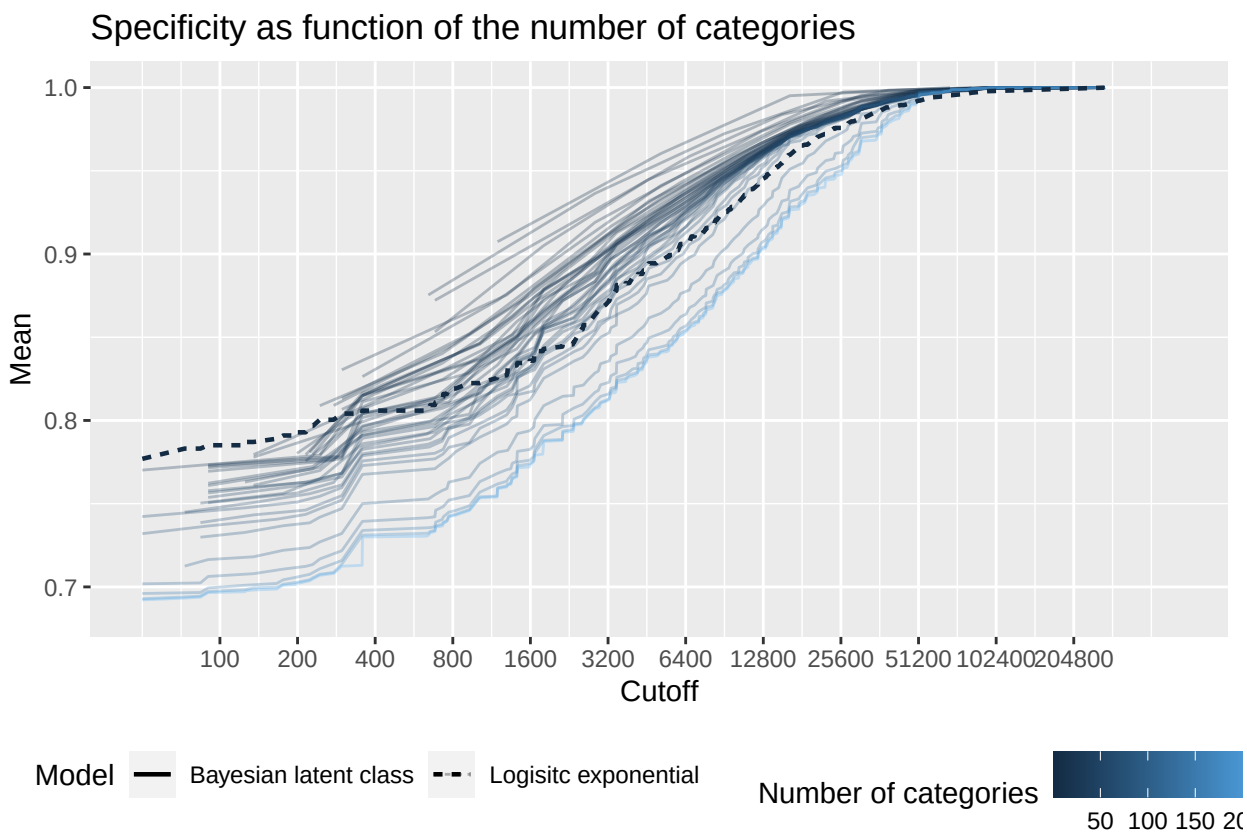
```
## Warning: Removed 6 row(s) containing missing values (geom_path).
```




```
# Plot of the specificites
sim_af_cats %>%
  filter(grepl("spec", varname)) %>%
  ggplot(
    aes(x = cutoff, y = Mean, group=ncuts, color = ncuts)) +
  geom_line(aes(linetype = "Bayesian latent class"), alpha = 0.3)+
  geom_line(data = dxp %>% filter(Indicator == "specificity") %>% mutate(ncuts = 1),
            aes(x = cutoff,
                y = Value,
                linetype = "Logisitc exponential",
                color = 1), size = 0.8 )+
  scale_x_log10("Cutoff", breaks=c(100,200,400,800,1600,3200,6400,12800,25600,51200,102400,204800), lim
  scale_linetype("Model") +
  scale_colour_continuous("Number of categories") +
  ggtitle("Specificity as function of the number of categories") +
  theme(legend.position = "bottom")
```

```
## Warning: Removed 74 row(s) containing missing values (geom_path).
```

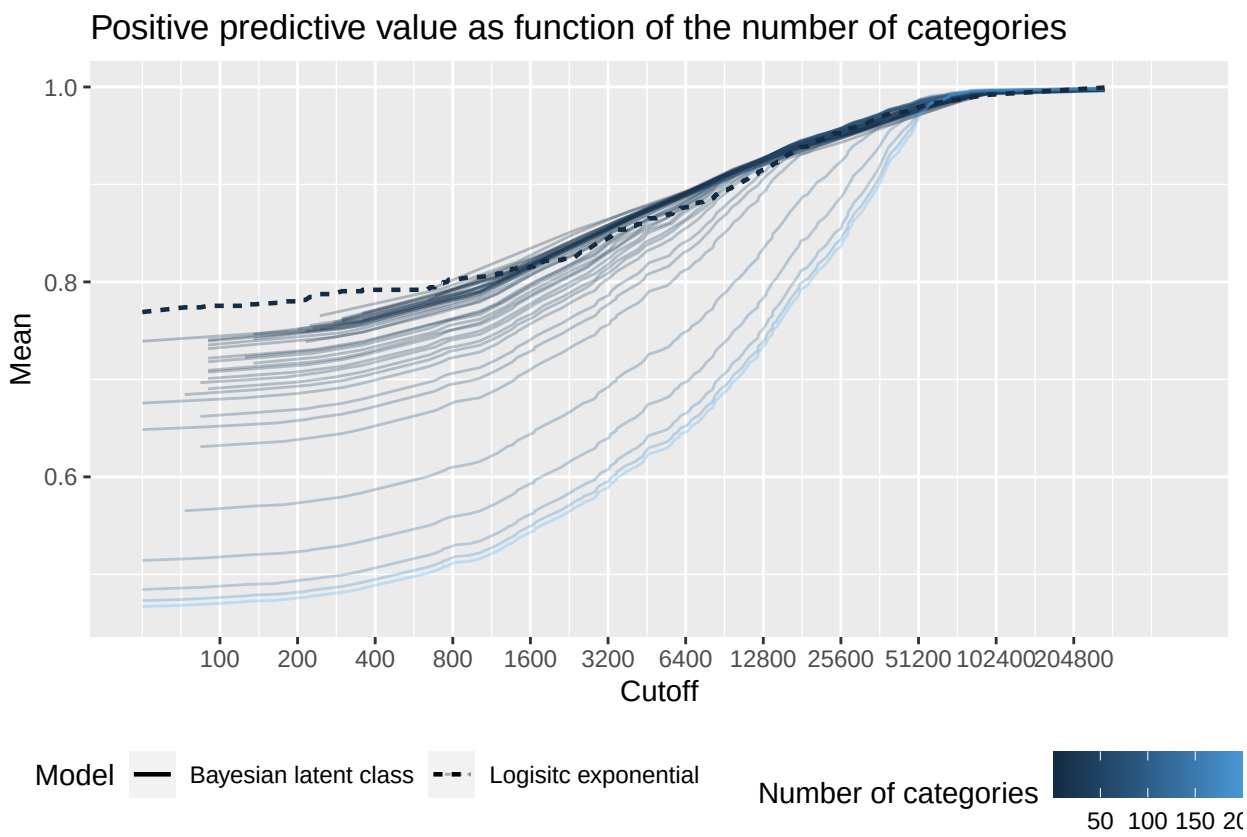
```
## Warning: Removed 6 row(s) containing missing values (geom_path).
```




```
# Plot of the ppv
sim_af_cats %>%
  filter(grepl("ppv", varname)) %>%
  ggplot(
    aes(x = cutoff, y = Mean, group=ncuts, color = ncuts)) +
  geom_line(aes(linetype = "Bayesian latent class"), alpha = 0.3)+
  geom_line(data = dxp %>% filter(Indicator == "ppv") %>% mutate(ncuts = 1),
            aes(x = cutoff,
                y = Value,
                linetype = "Logisitc exponential",
                color = 1), size = 0.8 )+
  scale_x_log10("Cutoff", breaks=c(100,200,400,800,1600,3200,6400,12800,25600,51200,102400,204800), lim
  scale_linetype("Model") +
  scale_colour_continuous("Number of categories") +
  ggtitle("Positive predictive value as function of the number of categories") +
  theme(legend.position = "bottom")
```

```
## Warning: Removed 74 row(s) containing missing values (geom_path).
```

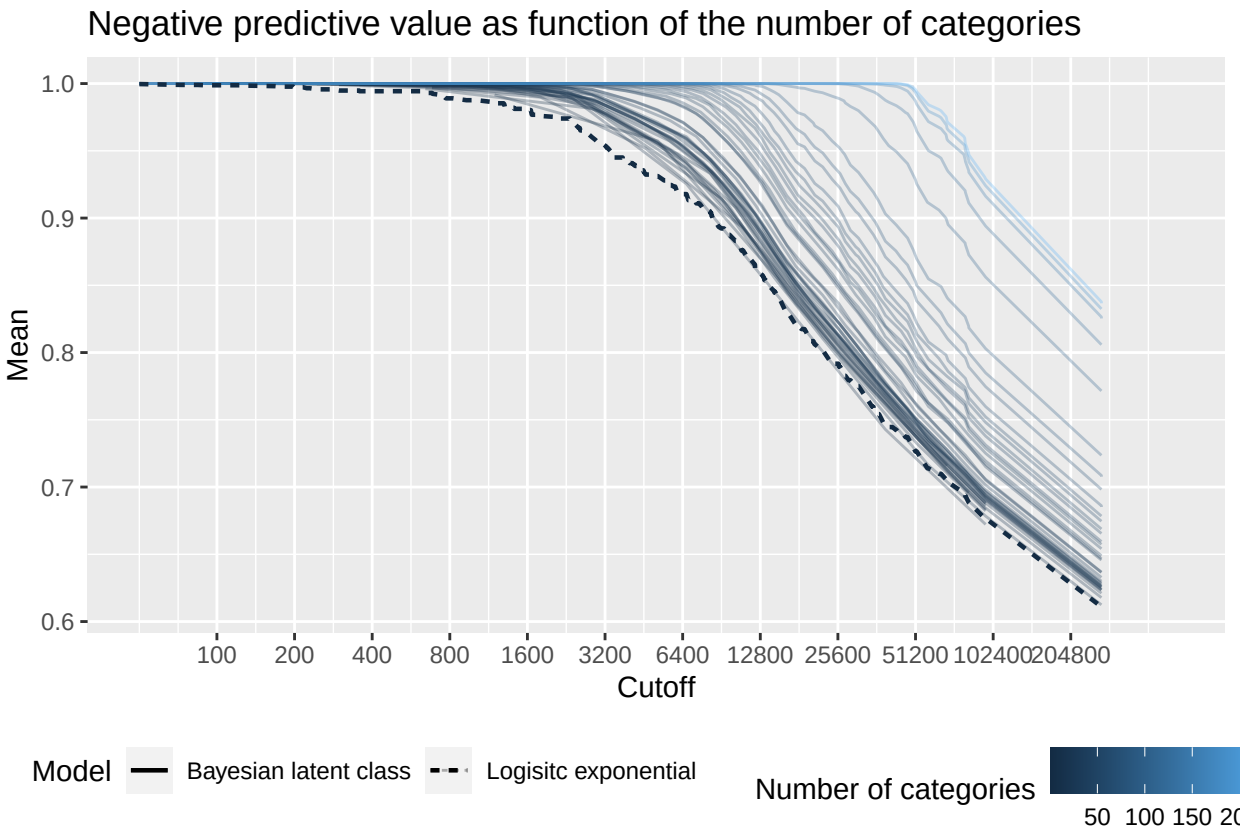
```
## Warning: Removed 6 row(s) containing missing values (geom_path).
```




```
# Plot of the npv
sim_af_cats %>%
  filter(grepl("npv", varname)) %>%
  ggplot(
    aes(x = cutoff, y = Mean, group=ncuts, color = ncuts)) +
  geom_line(aes(linetype = "Bayesian latent class"), alpha = 0.3)+
  geom_line(data = dxp %>% filter(Indicator == "npv") %>% mutate(ncuts = 1),
            aes(x = cutoff,
                y = Value,
                linetype = "Logisitc exponential",
                color = 1), size = 0.8 )+
  scale_x_log10("Cutoff", breaks=c(100,200,400,800,1600,3200,6400,12800,25600,51200,102400,204800), lim
  scale_linetype("Model") +
  scale_colour_continuous("Number of categories") +
  ggtitle("Negative predictive value as function of the number of categories") +
  theme(legend.position = "bottom")
```

```
## Warning: Removed 74 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 6 row(s) containing missing values (geom_path).
```


Negative predictive value as function of the number of categories
1.0
0.9
0.8
0.7
0.6
Mean
100
200
400
800
1600
3200
6400
12800
25600
51200
102400
204800
Cutoff
Model
Bayesian latent class
Logisitc exponential
Number of categories
50
100
150