PREDICT model checking

Nathan Green (Imperial College London

28/02/2020

```
library(readr)
library(R2jags)
library(R2WinBUGS)
library(purrr)
library(dplyr)
library(forcats)
library(lattice)
```

```
load(file = here::here("docs", "PREDICT main", "dat.RData"))              # design array with strings
load(file = here::here("docs", "PREDICT main", "jags_dat_input.RData"))   # jags input
load(file = here::here("data output", "out_regn.RData"))                   # jags output
```

Tables

```
# extract just prevalence results
prev_rows <- grepl("p_", rownames(out$BUGSoutput$summary))
p_latent <- out$BUGSoutput$summary[prev_rows, ]

# join with groups names
p_latent <-
  data.frame(dat[ ,c("age_grp",
                     "ethnicity",
                     "inc_cob_participant2",
                     "reasonforscreening")],
             mean = round(p_latent[,"mean"], 3),
             sd = round(p_latent[,"sd"], 3), row.names = NULL)

# head(p_latent)
knitr::kable(p_latent)

write.csv(p_latent, file = "../../data output/p_latent.csv")
```

```
# extract just prevalence results
prev_rows <- grepl("pp", rownames(out$BUGSoutput$summary))
p_latent <- out$BUGSoutput$summary[prev_rows, ]

# match up codes (levels) with names
lookup <-
  data.frame(
    rfs = 1:nlevels(dat$reasonforscreening) - 1,
    reasonforscreening = levels(dat$reasonforscreening))

# join with groups names
p_latent <-
```

```
  data.frame(dat_regn[ ,"rfs"],
             mean = round(p_latent[ ,"mean"], 3),
             sd = round(p_latent[ ,"sd"], 3),
             row.names = NULL) %>%
  merge(lookup, .)

# head(p_latent)
knitr::kable(p_latent)

write.csv(p_latent, file = "../../data output/ppred.csv")
```

```
# regression model: all covariates

# make sure these match up with ppred
# i.e. is the order the same as jags output
all_groups <-
  expand.grid(rfs = levels(dat$reasonforscreening),
              inc = levels(dat$inc_cob_participant2),
              eth = levels(dat$ethnicity),
              age = levels(dat$age_grp),
              bcg = levels(dat$prevbcg),
              yse = levels(dat$yearssinceentry_grp)
  )

BUGSsummary <- out$BUGSoutput$summary

# extract just prevalence results
prev_rows <- grepl("pp", rownames(BUGSsummary))
p_latent <- BUGSsummary[prev_rows, ]

# match up codes (levels) with names
lookup_rfs <-
  data.frame(rfs = 1:nlevels(dat$reasonforscreening) - 1,
             reasonforscreening = levels(dat$reasonforscreening))
lookup_inc <-
  data.frame(inc = 1:nlevels(dat$inc_cob_participant2),
             inc_cob_participant2 = levels(dat$inc_cob_participant2))
lookup_eth <-
  data.frame(eth = 1:nlevels(dat$ethnicity),
             ethnicity = levels(dat$ethnicity))
lookup_age <-
  data.frame(age = 1:nlevels(dat$age_grp),
             age_grp = levels(dat$age_grp))
lookup_bcg <-
  data.frame(bcg = 1:nlevels(dat$prevbcg),
             prevbcg = levels(dat$prevbcg))
lookup_yse <-
  data.frame(yse = 1:nlevels(dat$yearssinceentry_grp),
             yearssinceentry_grp = levels(dat$yearssinceentry_grp))


##TODO: this is a more robust approach
## determine programatically
```

```r
#
# group indicies
# gsub(pattern = "ppred", "", rownames(p_latent)) %>%
#   gsub("\\[", "", .) %>%
#   gsub("\\]", "", .) %>%
#   strsplit(",") %>%
#   do.call(rbind, .)

# join with groups names
# latent_tab <-
#   data.frame(dat_regn[ ,c("rfs", "inc", "eth", "age")]) %>%
#   arrange(age, eth, inc, rfs) %>%
#   mutate(mean = round(p_latent[ ,"mean"], 3),
#          sd = round(p_latent[ ,"sd"], 3),
#          row.names = NULL) %>%
#   merge(lookup_rfs, ., by = "rfs") %>%
#   merge(lookup_inc, ., by = "inc") %>%
#   merge(lookup_eth, ., by = "eth") %>%
#   merge(lookup_age, ., by = "age") %>%
#   arrange(age, eth, inc, rfs)


latent_tab <-
  all_groups %>%
  mutate(mean = round(p_latent[ ,"mean"], 3),
         sd = round(p_latent[ ,"sd"], 3),
         # index = rownames(p_latent),  #check matches with jags code
         row.names = NULL)

head(latent_tab) %>% knitr::kable()
```

rfs	inc	eth	age	bcg	yse	mean	sd
Contact	<40	White	(15,35]	No	(0,5]	0.235	0.027
Migrant	<40	White	(15,35]	No	(0,5]	0.171	0.031
Contact	41-100	White	(15,35]	No	(0,5]	0.322	0.031
Migrant	41-100	White	(15,35]	No	(0,5]	0.241	0.035
Contact	100-300	White	(15,35]	No	(0,5]	0.451	0.028
Migrant	100-300	White	(15,35]	No	(0,5]	0.354	0.040

```r
write.csv(latent_tab, file = here::here("data output", "ppred.csv"))
```

```r
other_rows <- rownames(BUGSsummary) %in% c("lambda", "sens", "spec")
othersummary <- BUGSsummary[other_rows, ]

other_tab <-
  othersummary %>%
  as.data.frame() %>%
  transmute(param = rownames(.),
            mean = round(mean, 3),
            sd = round(sd, 3),
            row.names = NULL)
```

```r
knitr::kable(other_tab)
```

param	mean	sd
lambda	0.040	0.004
sens	0.739	0.032
spec	0.990	0.005

```r
write.csv(other_tab, file = here::here("data output", "other_tab.csv"))
```

Predictive checks

We use ideas from Gabry et al. (2017).

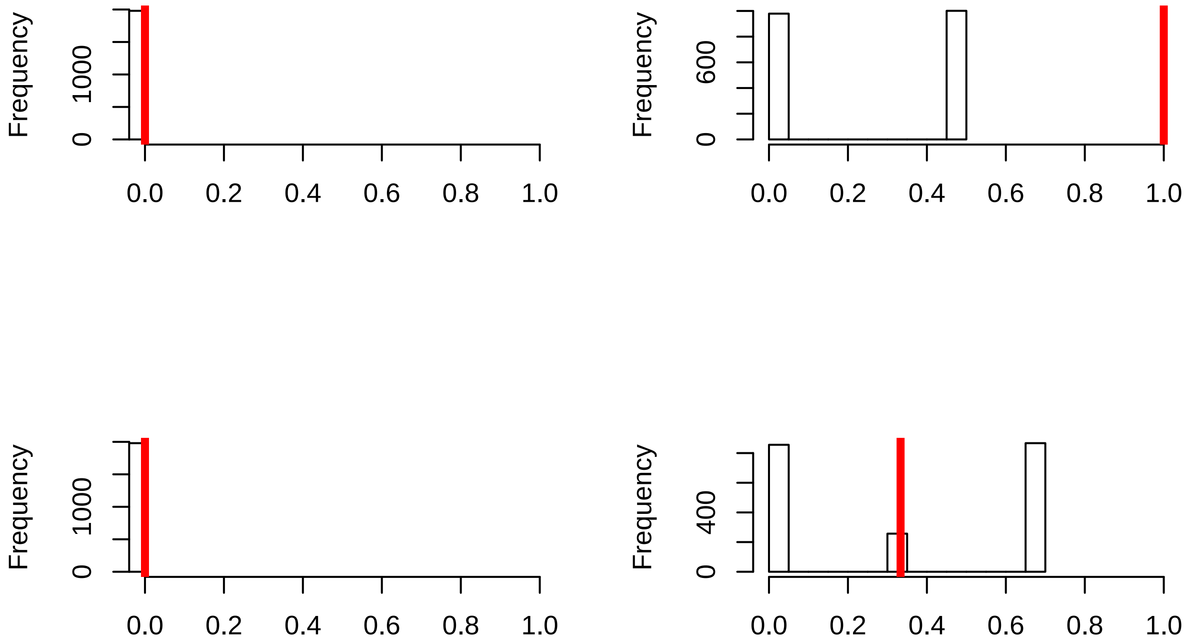
Prior predictive distributions

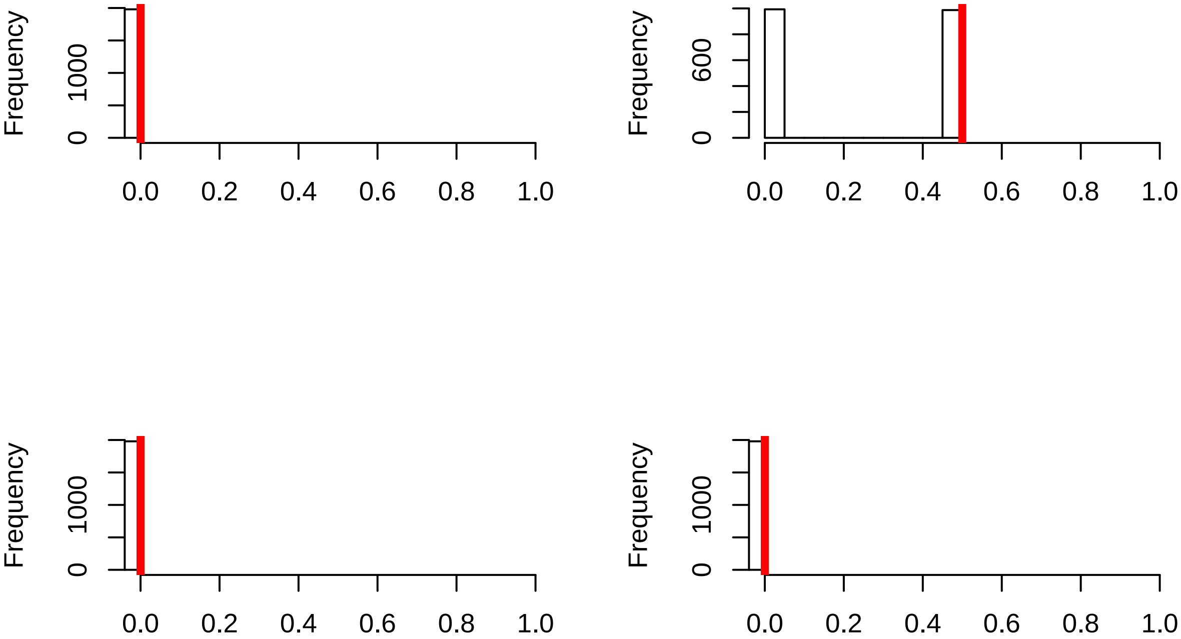
For LTBI there are when we use flat priors on the coefficients there are some very large counts generated from the prior. The prior produces a bowl-shaped distribution. To address this we use weakly informative priors.

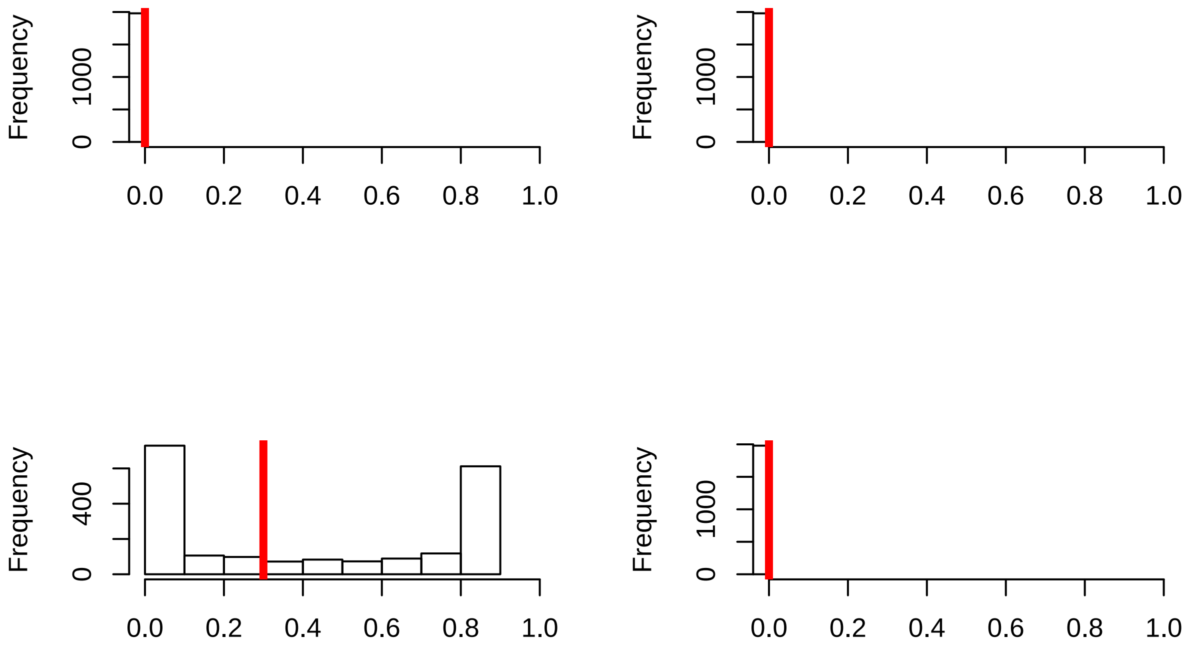
For a random selection of groups plot the prior predictive distributions and the observed data value.

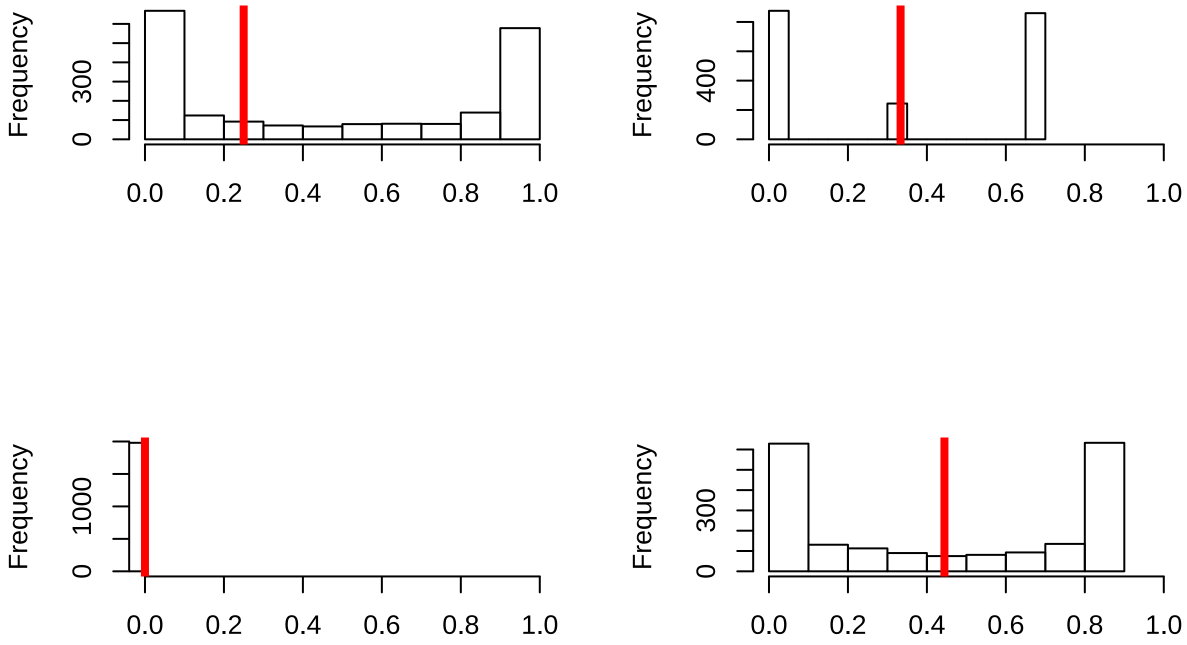
```r
par(mfrow = c(2,2))
for (i in 501:516) {

  hist(out$BUGSoutput$sims.list$prior_X_latent[, i]/jags_dat_input$Xm[i],
       main = "", xlab = "", xlim = c(0,1))
  abline(v = jags_dat_input$Xp[i]/jags_dat_input$Xm[i], col = "red", lwd = 4)
}
```


Frequency
0
1000
0.0
0.2
0.4
0.6
0.8
1.0
Frequency
0
600
0.0
0.2
0.4
0.6
0.8
1.0
Frequency
0
1000
0.0
0.2
0.4
0.6
0.8
1.0
Frequency
0
400
0.0
0.2
0.4
0.6
0.8
1.0

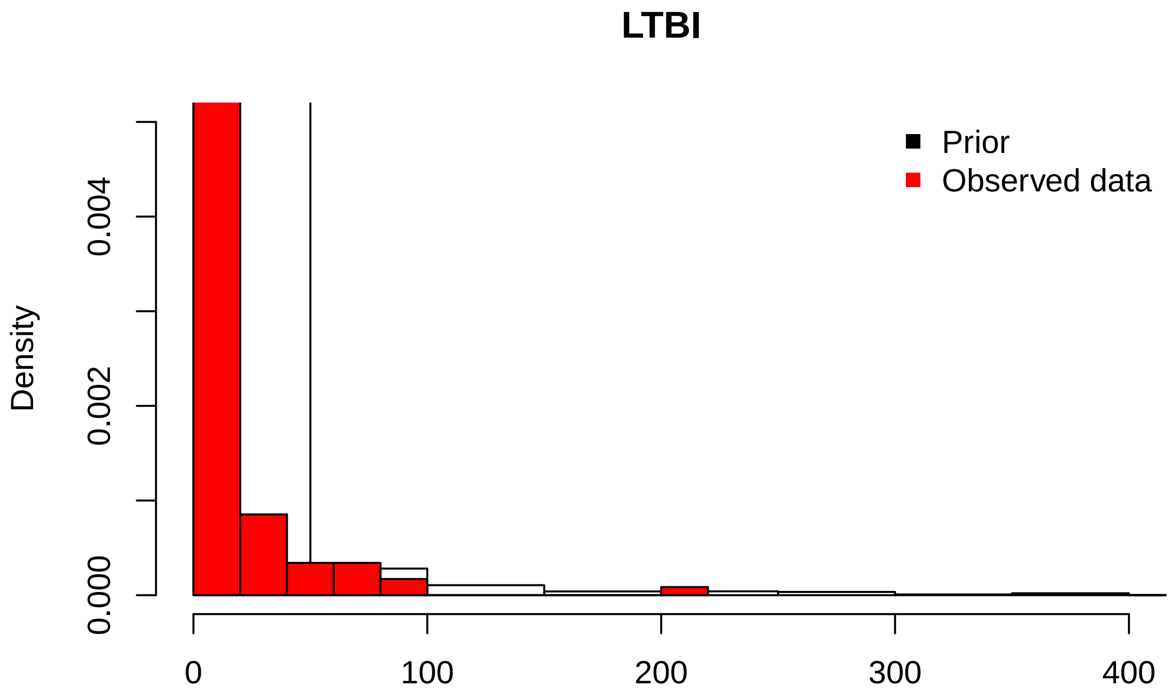

Frequency
0
1000
0.0
0.2
0.4
0.6
0.8
1.0
Frequency
0
600
0.0
0.2
0.4
0.6
0.8
1.0
Frequency
0
1000
0.0
0.2
0.4
0.6
0.8
1.0
Frequency
0
1000
0.0
0.2
0.4
0.6
0.8
1.0


Frequency
0
1000
0.0
0.2
0.4
0.6
0.8
1.0
Frequency
0
1000
0.0
0.2
0.4
0.6
0.8
1.0
Frequency
0
400
0.0
0.2
0.4
0.6
0.8
1.0
Frequency
0
1000
0.0
0.2
0.4
0.6
0.8
1.0



Pool across all groups.

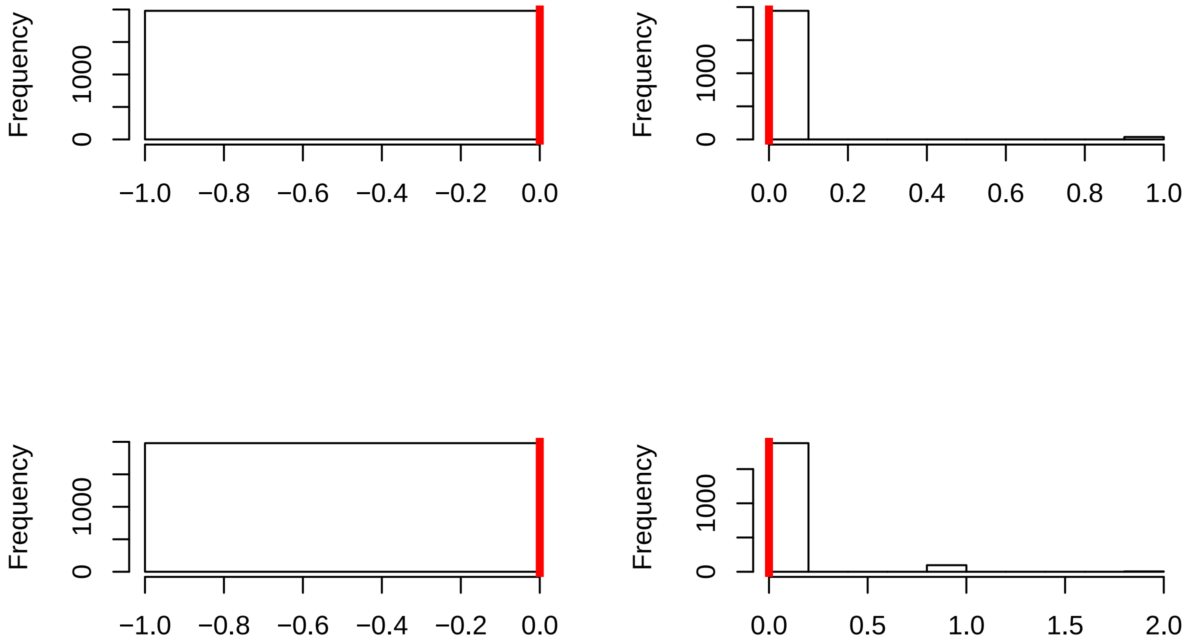
```r
hist(out$BUGSoutput$sims.list$prior_X_latent,
     main = "LTBI", freq = FALSE, xlab = "",
     ylim = c(0,0.005), xlim = c(0,400))
hist(jags_dat_input$Xp, freq = FALSE, col = "red", add = TRUE)
legend("topright", legend = c("Prior", "Observed data"),
       col = c("black","red"), pch = 15, bty = "n")
```

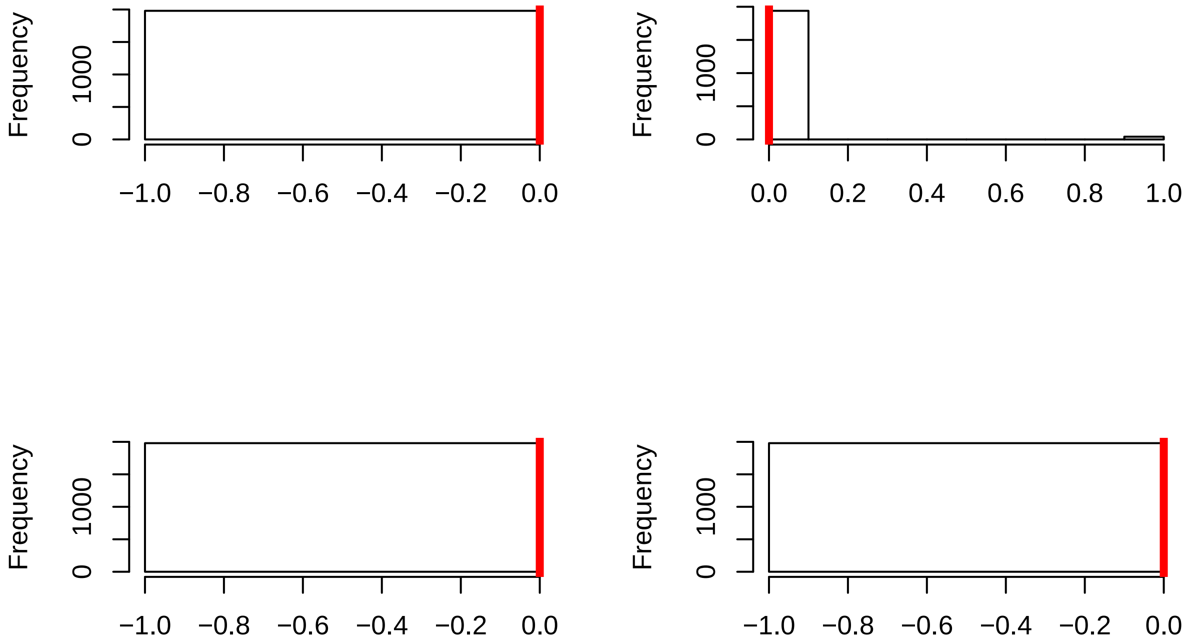


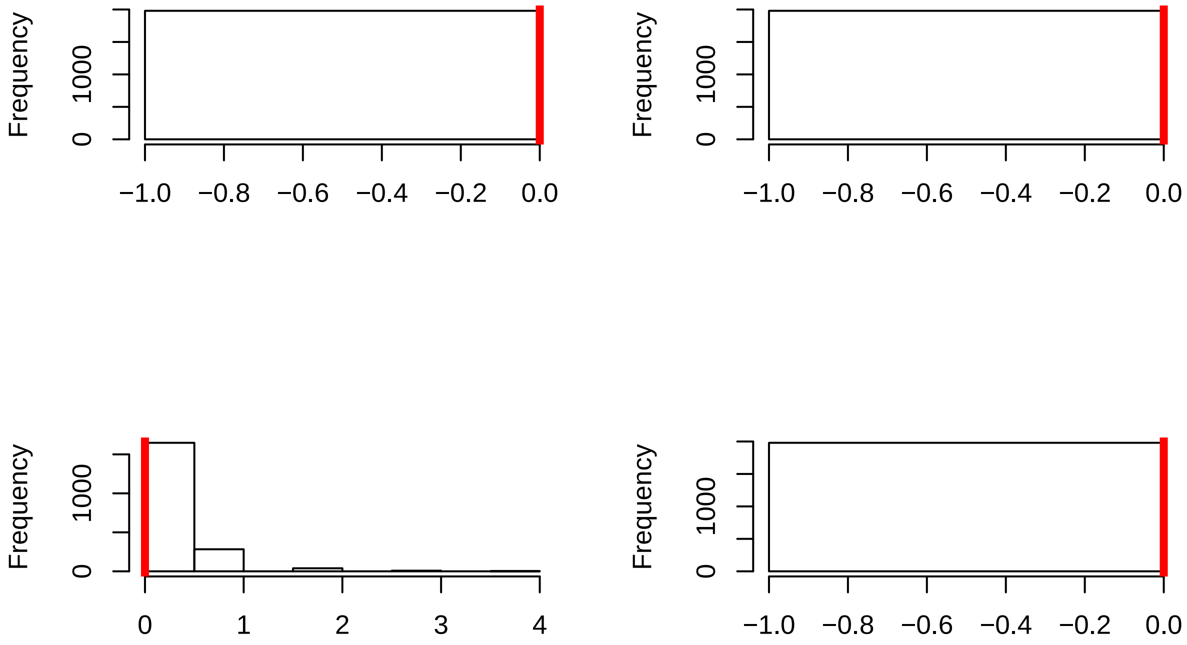
For TB, the distributions seem better in terms of capturing the data.

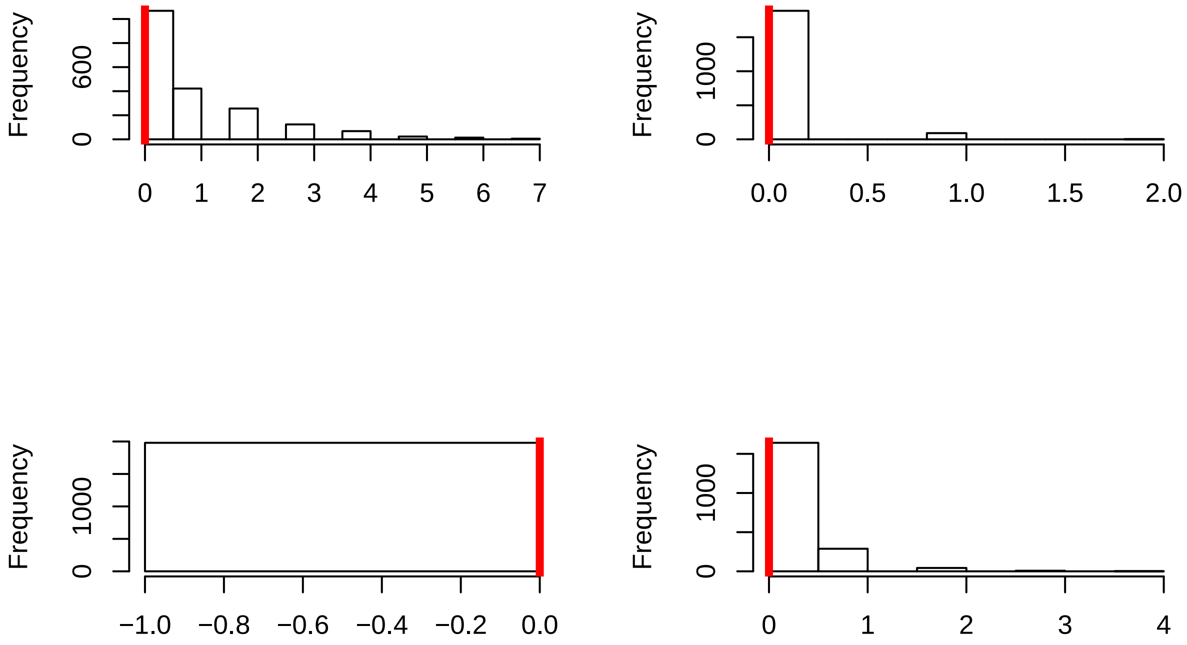
```
par(mfrow = c(2,2))
for (i in 501:516) {

  hist(out$BUGSoutput$sims.list$prior_Xtb[, i], main = "", xlab = "")
  abline(v = jags_dat_input$Xtb[1], col = "red", lwd = 4)
}
```

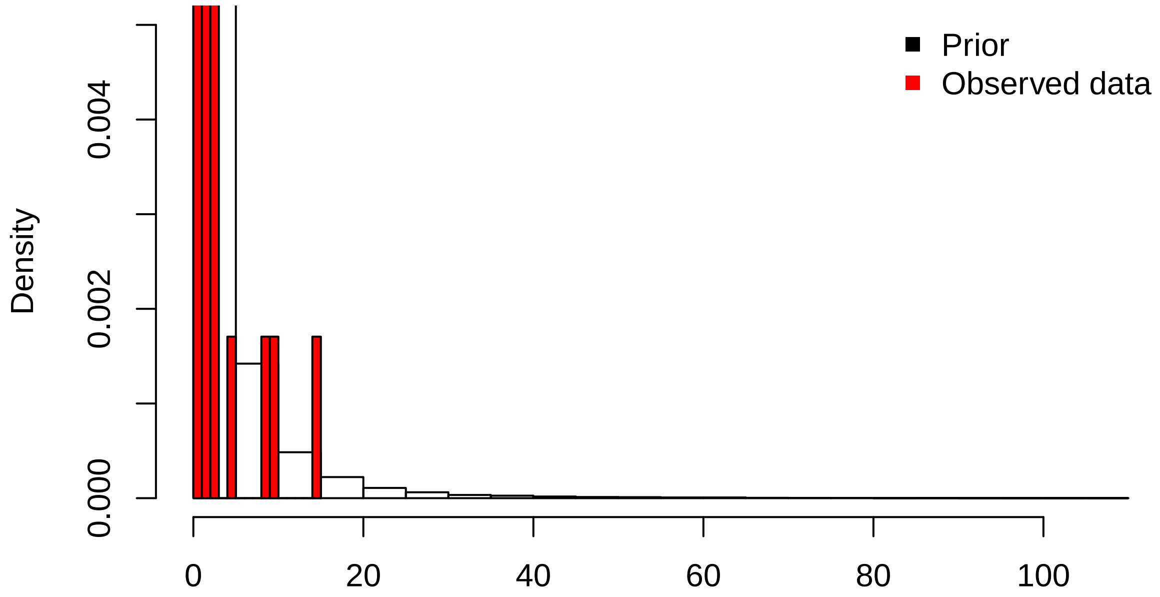

Frequency
0
1000
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
Frequency
0
1000
0.0
0.2
0.4
0.6
0.8
1.0
Frequency
0
1000
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
Frequency
0
1000
0.0
0.5
1.0
1.5
2.0


Frequency
0
1000
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
Frequency
0
1000
0.0
0.2
0.4
0.6
0.8
1.0
Frequency
0
1000
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
Frequency
0
1000
−1.0
−0.8
−0.6
−0.4
−0.2
0.0


Frequency
0
1000
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
Frequency
0
1000
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
Frequency
0
1000
0
1
2
3
4
Frequency
0
1000
−1.0
−0.8
−0.6
−0.4
−0.2
0.0



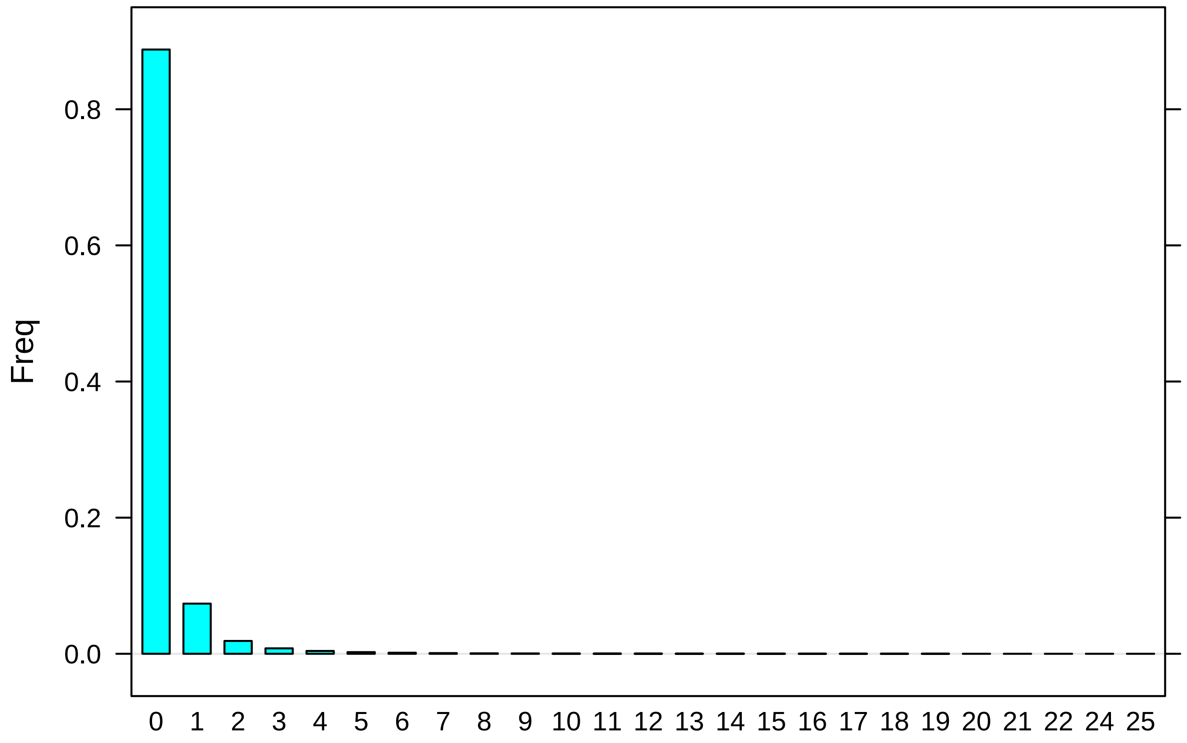
```
hist(out$BUGSoutput$sims.list$prior_Xtb,
     main = "", ylim = c(0,0.005), freq = FALSE, xlab = "")
hist(jags_dat_input$Xtb, freq = FALSE, col = "red", add = TRUE)
legend("topright", legend = c("Prior", "Observed data"),
       col = c("black","red"), pch = 15, bty = "n")
```



Posterior predictive distributions

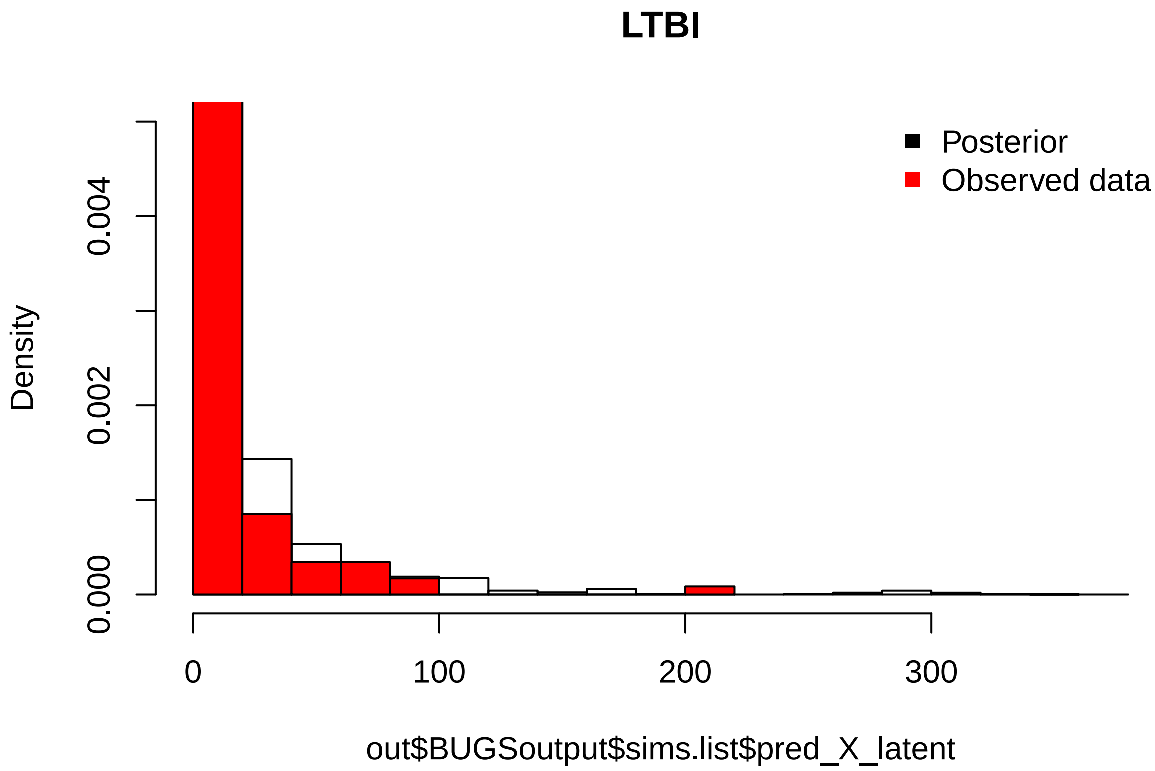
Check the posterior simulated data. Pick a single category of individual. Generate number of active TB cases per 100 individuals.

```r
out$BUGSoutput$sims.list$pred_Xtb %>%
  table() %>%
  prop.table() %>%
  barchart(horizontal = FALSE)
```

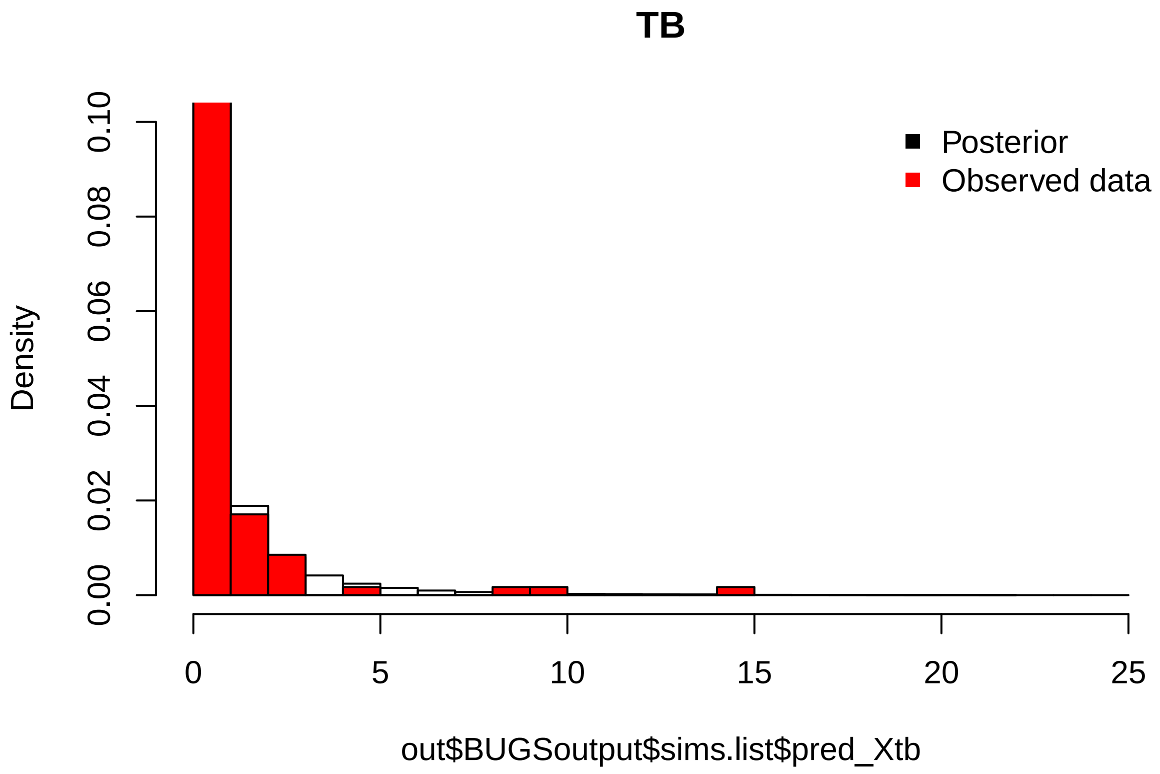


```
# indiv <- map_df(jags_dat_input, 1)
# indiv$Xtb/indiv$Xm

# pooled
hist(out$BUGSoutput$sims.list$pred_X_latent,
     main = "LTBI", ylim = c(0,0.005), freq = FALSE)
hist(jags_dat_input$Xp, freq = FALSE, col = "red", add = TRUE)
legend("topright", legend = c("Posterior", "Observed data"),
       col = c("black","red"), pch = 15, bty = "n")
```



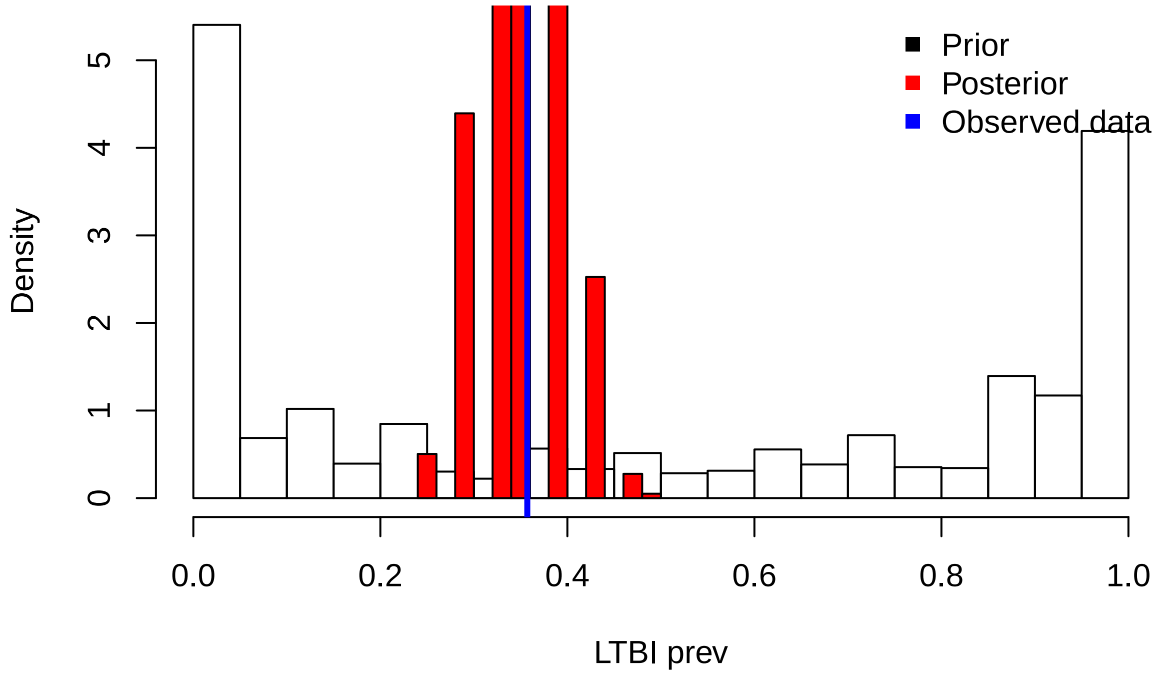
```r
hist(out$BUGSoutput$sims.list$pred_Xtb,
     main = "TB", ylim = c(0,0.1), freq = FALSE)
hist(jags_dat_input$Xtb, freq = FALSE, col = "red", add = TRUE)
legend("topright", legend = c("Posterior", "Observed data"),
       col = c("black","red"), pch = 15, bty = "n")
```



For a single group check prior, posterior and observed data.

```
# plot(density(out$BUGSoutput$sims.list$prior_X_latent[, 1], bw = 10, from = 0),
#      main = "LTBI", freq = FALSE)
# lines(density(out$BUGSoutput$sims.list$pred_X_latent, bw = 10, from = 0),
#      freq = FALSE, col = "red")

i <- 229
hist(out$BUGSoutput$sims.list$prior_X_latent[, i]/jags_dat_input$Xm[i],
     freq = FALSE,
     xlim = c(0,1),
     main = "",
     xlab = "LTBI prev", breaks = 20)
hist(out$BUGSoutput$sims.list$pred_X_latent[, i]/jags_dat_input$Xm[i],
     freq = FALSE, add = TRUE, col = "red")
abline(v = jags_dat_input$Xp[i]/jags_dat_input$Xm[i], lwd = 3, col = "blue")
legend("topright", legend = c("Prior", "Posterior", "Observed data"),
       col = c("black","red", "blue"), pch = 15, bty = "n")
```



Prior sensitivity analysis

Gelman et al. (2008)

Following Gelman et al. (2008), we first standardised the input variables by shifting them to have mean 0 and standard deviation 0.5. Lets use the data with `rfs` only to demonstrate.

```
dat_regn$rfs_std <- c(dat_regn$pop[2]/sum(dat_regn$pop),
                      - dat_regn$pop[1]/sum(dat_regn$pop))

# dat_regn$bcg_std <- dat_regn$rfs_std

dat_regn
```

We assume a student-t Cauchy distributions on the binary variables as recommended by Gelman et al. (2008).

Pareek et al. (2011)

We used the odds ratio statistics from Pareek et al. (2011) to estimate semi-informative priors with sensible ranges of values.

The grouping in Pareek are different to that used here. However, if we assume that the OR between groups are similar then we can obtain some orders of magnitute for the priors.

For example, for the sex variable with female as baseline, the adjusted OR is 1.3 (95% CI 1.0-1.8), which is $\beta = exp(1.3)$. So $\sigma^2 = ((exp(1.8) - exp(1.3))/1.96)^2 = 1.4749245$.

Therefore, we could set conservative but informative prior distributions of $N(0, 2)$ on the coefficients. In fact, variance 2.71 correspond to a uniform probaility (see BUGS book Lunn et al. (n.d.)) so we used this.

References

Gabry, Jonah, Daniel Simpson, Aki Vehtari, Michael Betancourt, and Andrew Gelman. 2017. "Visualization in Bayesian workflow," 389–402. https://doi.org/10.1109/NAFIPS.2004.1337440.

Gelman, Andrew, Aleks Jakulin, Maria Grazia Pittau, and Yu Sung Su. 2008. "A weakly informative default prior distribution for logistic and other regression models." *Annals of Applied Statistics* 2 (4): 1360–83. https://doi.org/10.1214/08-AOAS191.

Lunn, D J, Christopher H. Jackson, Nicky Best, and D. J. Spiegelhalter. n.d. *The BUGS book*.

Pareek, Manish, John P Watson, L Peter Ormerod, Onn Min Kon, Gerrit Woltmann, Peter J. White, Ibrahim Abubakar, and Ajit Lalvani. 2011. "Screening of immigrants in the UK for imported latent tuberculosis : a multicentre cohort study and cost-effectiveness analysis." *The Lancet Infectious Diseases* 11 (6): 435–44. https://doi.org/10.1016/S1473-3099(11)70069-X.