HAZ in Uganda

This report contains the R code and output of an analysis conducted on HAZ with DHS data from Uganda

---

```r
# Load all packages needed for the analysis
if (!require("pacman")) install.packages("pacman")
pkgs = c("sf", "dplyr", "PrevMap", "ggplot2", "tmap")
pacman::p_load(pkgs, character.only = T)

# Load external functions
source("R/functions.R")
```

Data cleaning

Coordinates were converted from lat/long WGS84 to UTM zone 35N (in Km). 10 clusters have been removed from the analysis because lat and long was not available. In the original dataset the HAZ ranges from a minimum of -6 to a maximum of 99.98. A total of 22 individuals were removed because an extreme value of HAZ was reporterd (equal to 99.98). Other 26 individuals have been removed because either HAZ or vitamin A was not recorded.

```r
# Load HAZ data
haz <- readr::read_csv("data/stunting_ug.csv")

# Remove missing LAT LONG (some entries have 0 values)
# Remove also non admissable values for HAZ (== 99.98 and missing values)
# Remove individuals with missing vitamin A
haz <- haz %>%
  filter(LONGNUM != 0, !is.na(HAZ), HAZ < 50, !is.na(VitaminA_microgram_per_Ml))

# Convert the LAT LONG coordinates to UTM (km) EPSSG: 32365
crs_utm_km <- epsgKM(32635)
haz_sp <- haz %>%
  st_as_sf(coords = c("LONGNUM", "LATNUM"), crs = 4326) %>%
  st_transform(crs = crs_utm_km)

# For a quick interactive view of the data run the following line of code
# mapview::mapview(haz_sp)

# Add two new columns with the coordinates in UTM (km)
haz[, c("utm_x", "utm_y")] <- st_coordinates(haz_sp)

# Transform vitamin A to the log scale
haz$log_VITA <- log(haz$VitaminA_microgram_per_Ml)
```

```r
# Select only the columns relevant for the anlysis
haz <- haz %>%
  dplyr::select(HAZ, log_VITA, agem = Age_Month,
                Cluster_ID, utm_x, utm_y)

# Check if there are any missing values
missing <- sapply(haz, function(x) sum(is.na(x)))
knitr::kable(missing, col.names = "Number of missing")
```

	Number of missing
HAZ	0
log_VITA	0
agem	0
Cluster_ID	0
utm_x	0
utm_y	0

Raster covariates

```r
# Load shapefile for Uganda
uga <- st_read("data/geodata/gadm36_UGA.gpkg", layer = "gadm36_UGA_0")
```

```
## Reading layer ‘gadm36_UGA_0’ from data source ‘/home/claudio/Dropbox/chicas/HAZ_uganda/data/g
## Simple feature collection with 1 feature and 2 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:           xmin: 29.5715 ymin: -1.48214 xmax: 35.00027 ymax: 4.234466
## epsg (SRID):    4326
## proj4string:    +proj=longlat +datum=WGS84 +no_defs
```

```r
# Convert to the reference system of the points
uga <- st_transform(uga, crs = crs_utm_km)

# Load population raster
pop <- raster("data/geodata/pop2016_100m.tif")

# Create prediction grid
pred <- create_grid(resolution = 5, study_area = uga, pop = pop,
                    cutoff = 0)

# Load raster covariates and align them to the prediction grid
elevation <- raster("data/geodata/elevation2000_100m.tif")
slope <- raster("data/geodata/slope2000_100m.tif")
evi <- raster("data/geodata/evi2016_1km.tif")
```

```r
elevation <- align_raster(pred = pred$raster, cov = elevation)
slope <- align_raster(pred = pred$raster, cov = slope)
evi <- align_raster(pred = pred$raster, cov = evi)

# Stack all the raster covariates together
covariates <- stack(elevation, slope, evi)

# Extract them at the observed locations
cov_obs  <- extract(covariates, haz[, c("utm_x", "utm_y")])

# Fill NA
# idna <- which(is.na(cov_obs[, 1]))
# cov_obs[idna, ] <- extract(covariates, haz[idna, c("utm_x", "utm_y")],
#                            small = T, buffer = 5)

haz[, c("elevation", "slope", "evi")] <- cov_obs

# Remove NA
haz <- haz[!is.na(haz$elevation), ]
```

Relationship between HAZ and individual variables

Let's have a look at the relationship between HAZ scores and individual levle variables. If this relatioship turns out to be non-linear we will accomodate for it. From Figure 1 we can't really see a clear relationship between vitamin A and HAZ. For age, it looks like there is a decrease in HAZ until approximately 20 months and then no relationship.

```r
haz %>%
  dplyr::select(HAZ, log_VITA, agem) %>%
  tidyr::gather(key = "variable", value = "value", -HAZ) %>%
  ggplot(aes(x = value, y = HAZ)) +
  geom_point(shape = 21, fill = "black", alpha = .3, size = .5) +
  geom_smooth(method = "gam", formula = y ~ s(x), size = .5) +
  facet_wrap(~ variable, scales = "free",
             labeller = labeller(variable = function(x) c("Age (months)", "Vitamin A (log)"))) +
  labs(y = "HAZ") +
  theme_bw()
```

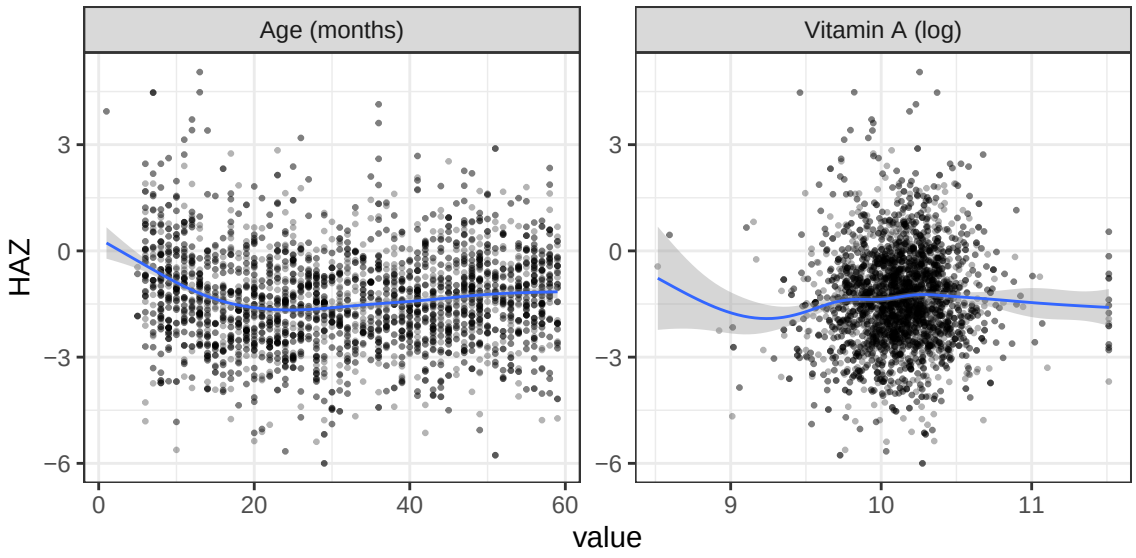


Figure 1: Scatterplot of the relationship between age (in month) and vitamin A (log scale) with HAZ. The blue line shows the fit of a GAM model (with a thin plate regression spline).

```
# Compare different models for HAZ vs. age
library(splines)
ggplot(haz, aes(x = agem, y = HAZ)) +
  geom_point(shape = 21, fill = "black", alpha = .3, size = 1) +
  geom_smooth(method = "gam", formula = y ~ s(x), aes(col = "GAM", fill = "GAM"),
              size = 1) +
  geom_smooth(method = "lm", formula = y ~ bs(x, knots = 17.84, degree = 1),
              aes(col = "LINEAR", fill = "LINEAR"), size = 1) +
  labs(x = "Age (months)", y = "HAZ") +
  scale_color_brewer("Model fit", type = "q", palette = 6) +
  scale_fill_brewer("Model fit", type = "q", palette = 6) +
  theme_bw()
```

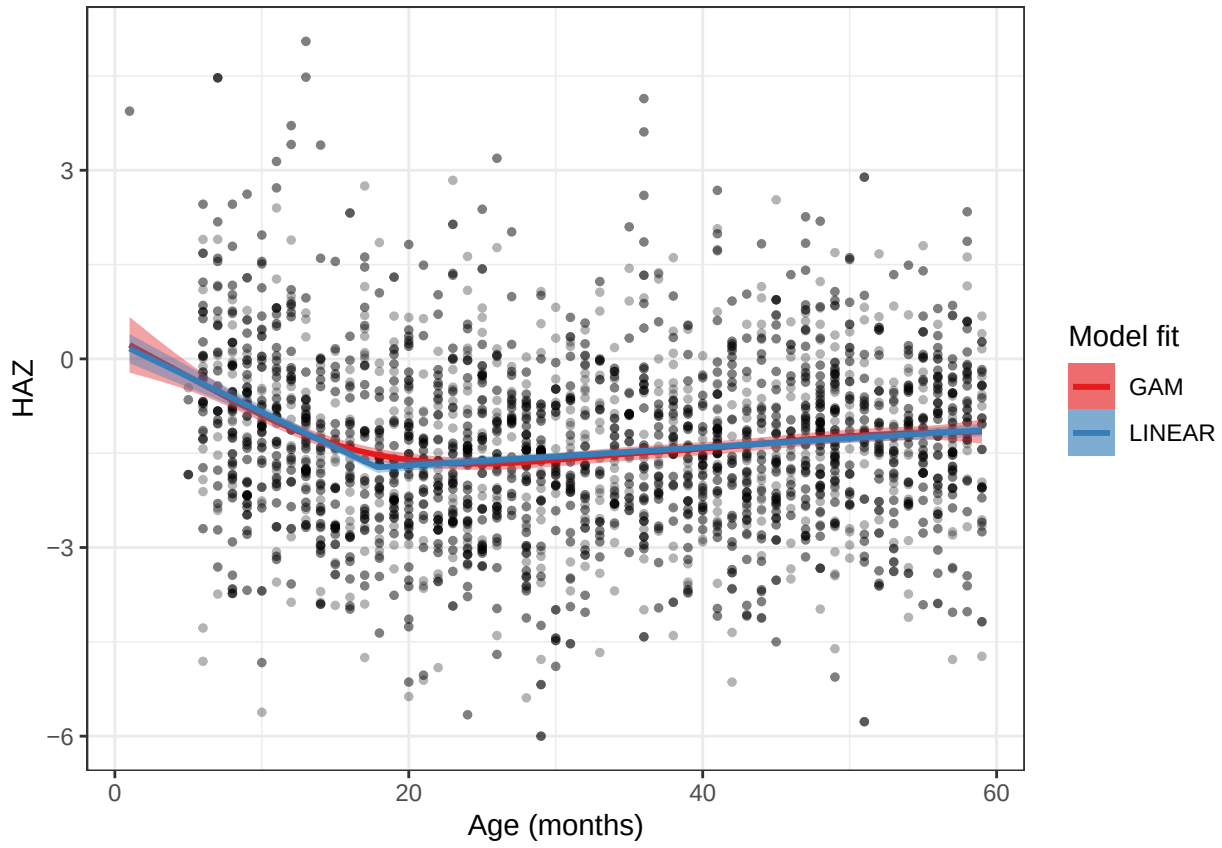


Figure 2: Comparison of the fit of a GAM and piece-wise linear regression to age and HAZ.

```
# Find optimal break point
# fcost <- function(param) {
#   fit <- lm(HAZ ~ bs(agem, knots = param, degree = 1), data = haz)
#   aic <- AIC(fit)
#   return(aic)
# }
#
# optim(par= 0, fn = fcost, method = "Brent", lower = -2, upper = 2)
```

Non-spatial model

Here we fit the following linear mixed model to the HAZ data

$$Y_j(x_i) = \alpha + \gamma d_{ij} + \beta d(x_i) + U_i + Z_{ij} \tag{1}$$

where $Y_j(x_i)$ denotes the HAZ of the jth child at cluster location x_i, d_{ij} is a vector of individual level covariates (age and vitamin A), $d(x_i)$ is the vector of spatially referenced environmental variables (elevation, slope and EVI). U_i and Z_{ij} are two set of independent normally distributed random effects who capture cluster level and individual level variation respectively. We use the estimated $\hat{U}_i$ to calculate the variogram and check the presence of residual spatial variation.

```r
# Load package to fit Mixed Model
library(lme4)

# Before fitting this type of models it is alway a good idea to rescale the
# numeric variables (substract the mean and divide by the standard devation)
# to avoid problem of convergence

# Scale the numeric covariates
# num_covariates <- c("agem", "log_VITA", "elevation", "slope", "evi")
# haz[, num_covariates] <- scale(haz[, num_covariates])

# Create new ID for clusters
haz <- as.data.frame(haz)
haz$ID <- create.ID.coords(data = haz, ~ utm_x + utm_y)
haz$Cluster_ID <- NULL

# Formula
f <- HAZ ~ agem + I((agem - 17.84) * (agem > 17.84)) + log_VITA +
  elevation + slope + evi

# Formula with cluster level random effects
fcluster <- update(f, ~ . + (1|ID))

# Fit the model in equation (1)
fit <- lmer(formula = fcluster, data = haz)

# Generate summary of the model
summary(fit)
```

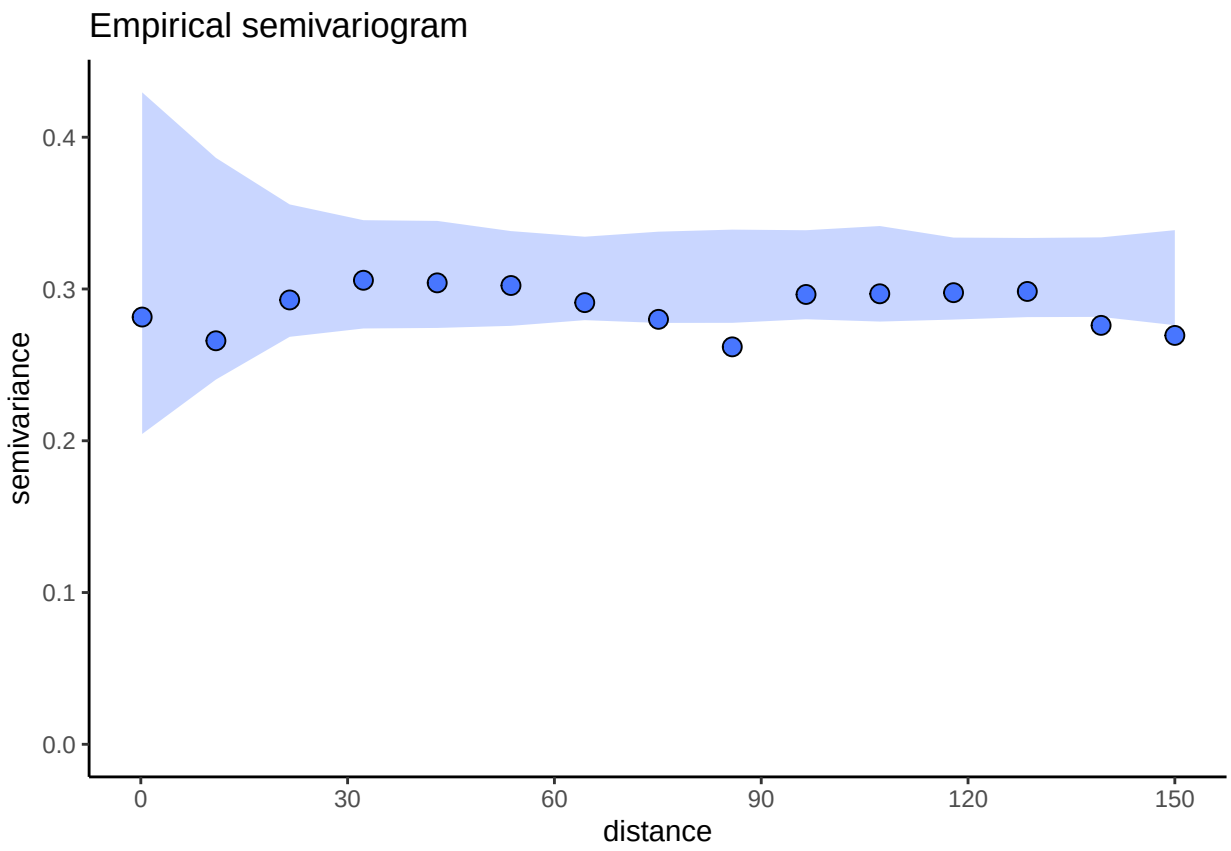
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: HAZ ~ agem + I((agem - 17.84) * (agem > 17.84)) + log_VITA +
##     elevation + slope + evi + (1 | ID)
##    Data: haz
##
## REML criterion at convergence: 12146.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2402 -0.6017 -0.0323  0.5392  4.3494
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ID       (Intercept) 0.5068   0.7119
##  Residual             1.3913   1.1795
## Number of obs: 3623, groups:  ID, 599
##
## Fixed effects:
```

```
##                                     Estimate Std. Error t value
## (Intercept)                        -9.074e-01  9.199e-01  -0.986
## agem                               -1.068e-01  7.969e-03 -13.407
## I((agem - 17.84) * (agem > 17.84))  1.209e-01  9.071e-03  13.328
## log_VITA                            1.499e-01  8.572e-02   1.748
## elevation                          -1.753e-05  2.525e-04  -0.069
## slope                              -4.246e-02  1.870e-02  -2.271
## evi                                -5.942e-01  4.936e-01  -1.204
##
## Correlation of Fixed Effects:
##             (Intr) agem   I-1*(>1 l_VITA elevtn slope
## agem        -0.126
## I((-17.*(>1  0.121 -0.986
## log_VITA    -0.929 -0.005  0.002
## elevation   -0.263  0.010 -0.009  -0.034
## slope        0.246 -0.014  0.012  -0.004 -0.759
## evi         -0.186  0.010 -0.010  -0.017  0.098 -0.331
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

```r
# Extract the random effects at cluster level U_i
reff <- ranef(fit)$ID$`(Intercept)`

# Extract coordinates for each cluster
coords <- haz %>%
  distinct(ID, utm_x, utm_y) %>%
  arrange(ID) %>%
  dplyr::select(utm_x, utm_y)

# Load ggvario function
ggvario(coords = coords, data = reff, nsim = 1000, show_nbins = F, maxdist = 150)
```



Geostatistical model

We improve equation (1) by replacing the cluster level random effects with a spatial gaussian process and we fit the following geostatistical model

$$Y_j(x_i) = \alpha + \gamma d_{ij} + \beta d(x_i) + S(x_i) + Z_{ij} \tag{2}$$

```
fit_geo <- linear.model.MLE(formula = f,
                            coords = ~ utm_x + utm_y,
                            ID.coords = haz$ID,
                            data = haz,
                            start.cov.pars = c(15, 0.2),
                            fixed.rel.nugget = 0, kappa = 0.5, method = "nlminb",
                            messages = F)
summary(fit_geo, l = F)
```

```
## Geostatistical linear model
## Call:
## linear.model.MLE(formula = f, coords = ~utm_x + utm_y, data = haz,
##     ID.coords = haz$ID, kappa = 0.5, fixed.rel.nugget = 0, start.cov.pars = c(15,
##         0.2), method = "nlminb", messages = F)
##
##                                      Estimate      StdErr  z.value
```

```
## (Intercept)                          -9.8061e-01  9.2504e-01  -1.0601
## agem                                 -1.0697e-01  7.9639e-03 -13.4320
## I((agem - 17.84) * (agem > 17.84))    1.2104e-01  9.0655e-03  13.3513
## log_VITA                              1.4999e-01  8.5703e-02   1.7501
## elevation                             8.4161e-06  2.5432e-04   0.0331
## slope                                -4.5142e-02  1.9029e-02  -2.3722
## evi                                  -4.6413e-01  5.2740e-01  -0.8800
##                                      p.value
## (Intercept)                          0.28911
## agem                                 < 2e-16 ***
## I((agem - 17.84) * (agem > 17.84)) < 2e-16 ***
## log_VITA                             0.08010 .
## elevation                            0.97360
## slope                                0.01768 *
## evi                                  0.37884
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-likelihood: -2719.623
##
## Covariance parameters Matern function
## (fixed relative variance tau^2/sigma^2= 0)
##         Estimate StdErr
## sigma^2  0.50264 0.1831
## phi      0.85266 0.8025
## omega^2  1.38995 0.2525
##
## Legend:
## sigma^2 = variance of the Gaussian process
## phi = scale of the spatial correlation
## omega^2 = variance of the individual unexplained variation
```

```
# Save the results
# saveRDS(fit_geo, file = "output/fit_geo_070220.rds")
```

Predictions

```
# Extract values of covariates at prediction locations
cov_pred <- extract(covariates, pred$coords)
cov_pred <- as.data.frame(cov_pred)
names(cov_pred) <- c("elevation", "slope", "evi")

cov_pred$agem <- mean(haz$agem)
cov_pred$log_VITA <- mean(haz$log_VITA)

# Generate distribution of predictors at individual level
```

```r
nsim <- 1000

# Create function to sample from age
sample_age <- function(nsim) {
  t.age <- haz$agem
  bw <- density(t.age)$bw
  t.age.obs <- t.age[sample(1:nrow(haz), nsim, replace = TRUE)]
  t.age.sim <- rnorm(n.sim,mean=t.age.obs,sd=bw)
  5*exp(t.age.sim)/(1+exp(t.age.sim))
}

# Obtain predictions
predictions <- spatial.pred.linear.MLE(fit_geo,
                                       grid.pred = pred$coords,
                                       predictors = cov_pred,
                                       n.sim.prev = nsim,
                                       scale.predictions = "logit",
                                       # type="joint",
                                       thresholds = -2,
                                       scale.thresholds = "logit")
```

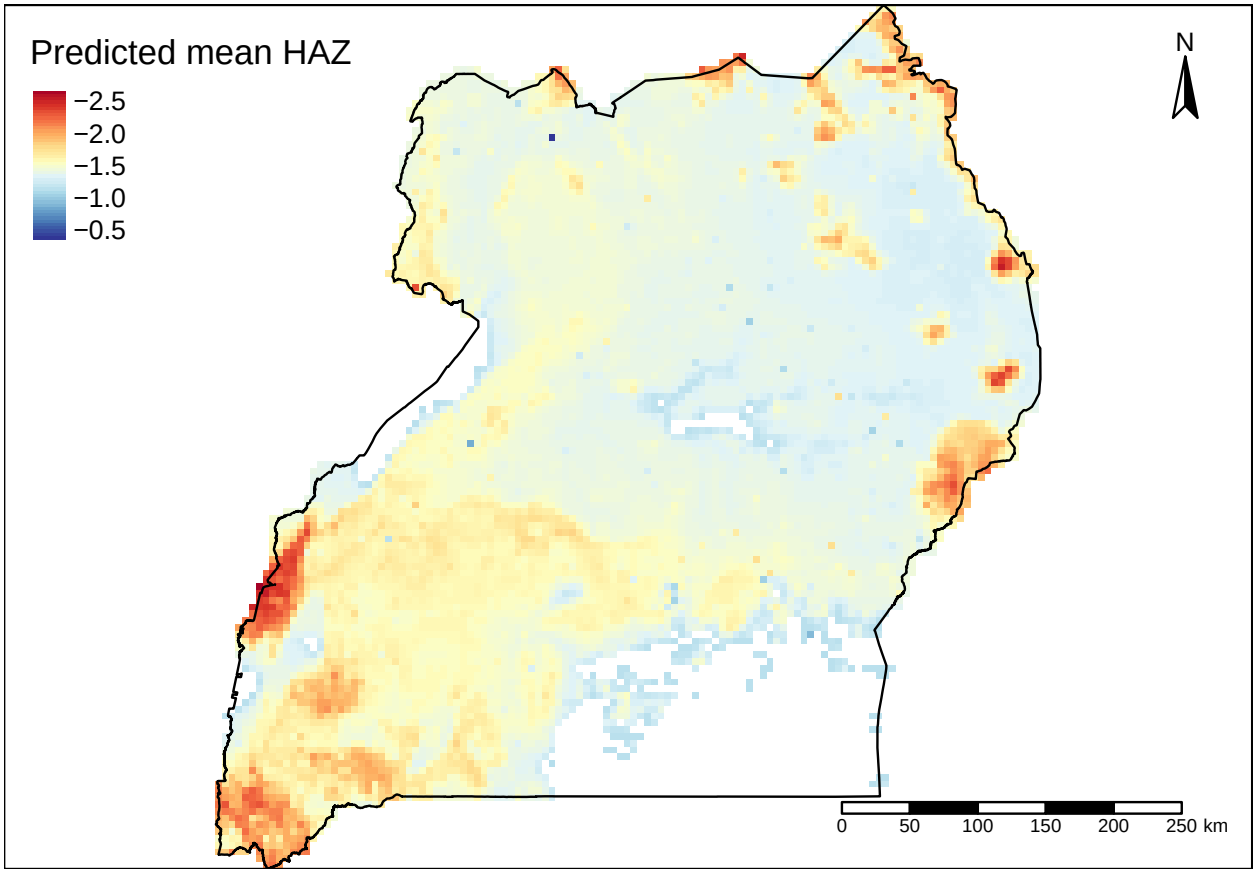
```
## NOTE: the nugget effect IS NOT included in the predictions.
## Type of prevalence predictions: marginal
## Spatial predictions: logit
```

```r
predictions$exceedance.prob <- 1 - predictions$exceedance.prob

# Save predictions
# saveRDS(predictions, "output/geo_pred_070220.rds")
```

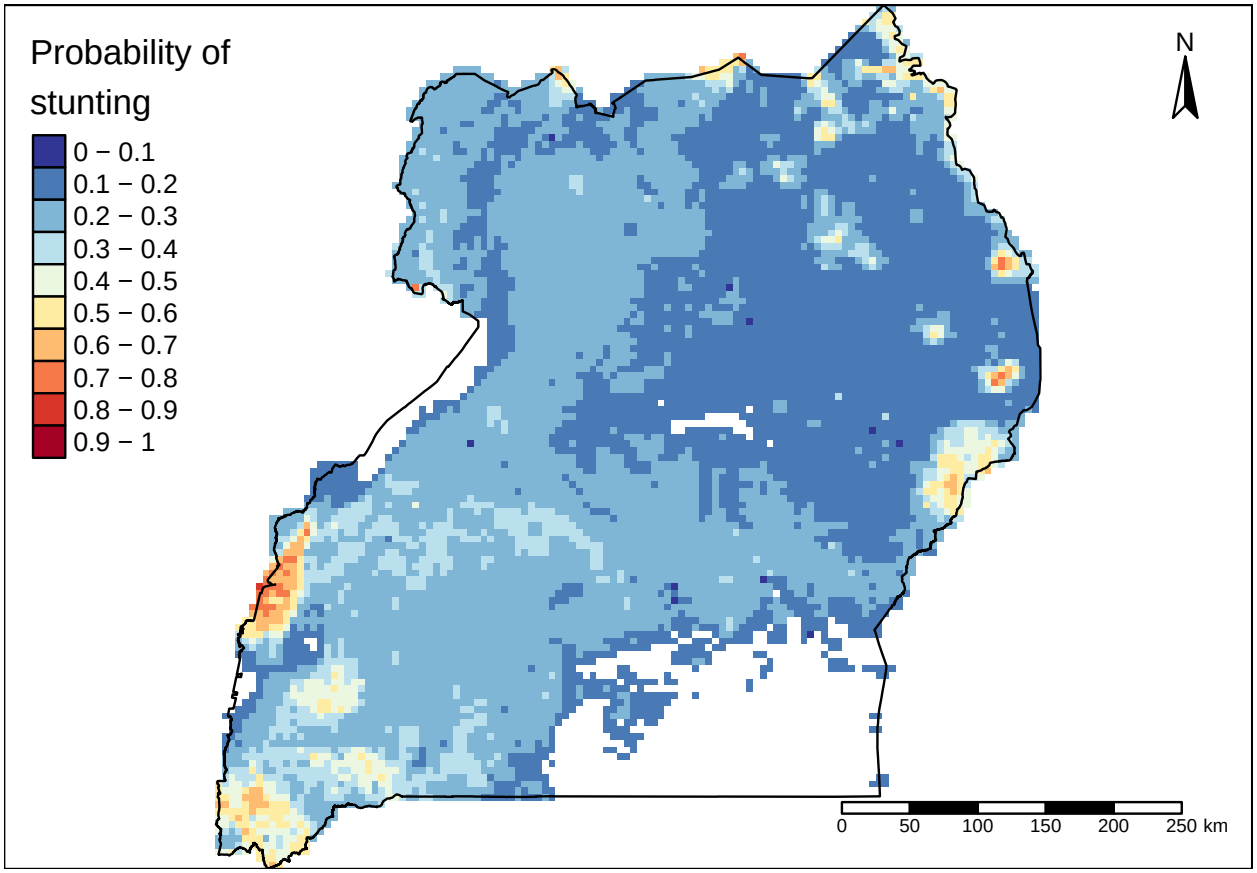
```r
haz_pred <- rasterFromXYZ(data.frame(pred$coords,
                                     haz = predictions$logit$predictions,
                                     pstunting = as.numeric(predictions$exceedance.prob)))

tm_shape(haz_pred) +
  tm_raster(col = "haz", palette = "RdYlBu", legend.show = T, style = "cont",
            title = "Predicted mean HAZ") +
tm_shape(uga, is.master = T,) +
  tm_borders("black", lwd = 1) +
  tm_compass(position = c("right", "top")) +
  tm_scale_bar(position = c("right", "bottom")) +
  tm_layout(design.mode = F, legend.bg.color = "white", scale = 1.2,
            legend.position = c("left", "top"), frame = T, outer.margins = 0, asp = 0)
```




```
pal_ex <- tmaptools::get_brewer_pal("-RdYlBu", n = 10, contrast = c(0, 1), plot = F)

tm_shape(haz_pred) +
  tm_raster(col = "pstunting", palette = pal_ex, legend.show = F,
            style = "fixed", breaks = seq(0, 1, by = .1)) +
  tm_add_legend(type = "fill",
                labels = c("0 - 0.1", "0.1 - 0.2", "0.2 - 0.3", "0.3 - 0.4", "0.4 - 0.5",
                           "0.5 - 0.6", "0.6 - 0.7", "0.7 - 0.8", "0.8 - 0.9", "0.9 - 1"),
                col = pal_ex, size = .5, alpha = 1,
                title = "Probability of\nstunting",
                is.portrait = T) +
tm_shape(uga, is.master = T,) +
  tm_borders("black", lwd = 1) +
  tm_compass(position = c("right", "top")) +
  tm_scale_bar(position = c("right", "bottom")) +
  tm_layout(design.mode = F, legend.bg.color = "white", scale = 1.2,
            legend.position = c("left", "top"), frame = T, outer.margins = 0, asp = 0)
```


Probability of stunting
0 − 0.1
0.1 − 0.2
0.2 − 0.3
0.3 − 0.4
0.4 − 0.5
0.5 − 0.6
0.6 − 0.7
0.7 − 0.8
0.8 − 0.9
0.9 − 1
N
0
50
100
150
200
250 km