BIOS0052: Human And Ecosystem Health In A Changing World

Week 4 practical session

Predicting the outbreak dynamics of dengue in central Vietnam

In today's practical we will use long-term dengue surveillance data from two provinces in Central Vietnam to develop predictive models of dengue outbreak dynamics. Like last week's practical, we will use climate data, spatial data tools (e.g. *terra* and *sf*) and the *mgcv* package for generalised additive modelling, and this week we will also use the *ranger* and *caret* packages to develop random forest (machine learning) models. Our objective is to explore a simple example of how predictive models are built and evaluated - in this case, exploring whether climate information can help to forecast unobserved dengue outbreaks.

Dengue is an acute and occasionally severe mosquito-borne viral disease - transmitted mainly by the mosquitoes *Aedes aegypti* and *Aedes albopictus* - whose incidence has surged in the last 30 years, to become one of the most significant infectious disease threats of tropical regions. Vietnam is one of the world's dengue hotspots, with endemic year-on-year transmission in most of the country, and growing incidence in cooler regions that may be linked to a warming climate. As we learned in last week's session, dengue is highly climate-sensitive due to the biology of its mosquito vectors, and in recent years there has been growing interest in using climate data to forecast outbreaks in advance, to aid in planning and preparedness. For more on the dengue situation and forecasting in Vietnam, you can read Gibb et al. 2023 and Colon et al. 2021, which provided the data we are using in today's practical.

This workbook is structured as a series of code snippets with short exercises interspersed. The solutions for the short exercises are available in the Rmarkdown script in the GitHub folder, but *please avoid looking at these* until you have tried to solve them!

At the end, there are some **longer extension exercises** to allow you to apply your knowledge and prepare you for the assessments. The solutions for these will be uploaded to Moodle next week.

All the data you'll need for the workshop are in the GitHub, in the "Week4-Prediction" folder. Please download the whole folder using download-directory.github.io. Set this folder as your working directory, then all the materials you will need are contained within the "data" subfolder.

```r
# package dependencies
# use the "install.packages()" command if not already installed
library(terra); library(dplyr); library(magrittr); library(ggplot2); library(sf)
library(rstudioapi); library(tidyr); library(stringr); library(mgcv); library(gratia)

# additional packages you may need to install
library(caret); library(ranger); library(pROC)

# automatically set working directory
# (or if this doesn't work,
# manually set your working directory to the folder "Week3-Measuring-Env-Effects")
PATH = dirname(rstudioapi::getSourceEditorContext()$path)
setwd(PATH)
```

Our research questions

The climate-sensitivity of dengue and other mosquito-borne viruses makes them potentially very good candidates for climate-driven prediction and forecasting. So our analyses today will focus on the question: **how accurately can we predict dengue outbreaks using climate information?**. We will use data from two provinces in Vietnam, Khanh Hoa (a coastal province with generally high dengue incidence) and Dak Lak (a highland province with generally lower and more variable dengue incidence). We will explore these datasets, develop and evaluate statistical predictive models using *mgcv*, then compare their performance to a machine learning (random forest) model which we will develop using the *caret* and *ranger* packages.

Exploring and understanding the dataset

The GitHub contains a dataset of **monthly dengue case counts**, collected between 2005 and 2021 at the district-level within Khanh Hoa and Dak Lak provinces, and compiled via Vietnam's national dengue surveillance system. The data were collected via passive (hospital-based) survillance of dengue cases, with diagnosis and reporting based on symptoms (i.e. syndromic diagnosis, so not all cases in the data were laboratory-confirmed). Each case count has a date and district associated with it, and has been assigned to a "dengue year" (which runs from May to April) rather than calendar year.

As well as the absolute case counts, the data frame also contains **two binary outbreak indicators, which define each month as an "outbreak" or not (1/0) depending on whether cases exceed a specified threshold**: either above the 95th percentile of historical counts prior to that date, or above the mean plus standard deviation of historical counts prior to that date. We will use these to model the relationship between climate and the probability of an outbreak.

The dataset also contains additional monthly district-level data on the population, the proportion of the population living in urban areas, and several climate indicators from ERA5: **average daily minimum temperature** (Tmin) and **precipitation** (mm) at lags of 1 and 2 months prior to reporting, and a **long-term drought indicator** (SPEI-6) at lags of 4 and 5 months prior to reporting. SPEI-6 (Standardised Precipitation Evapotranspiration Index) is a measure of long-term hydrometeorological excess or deficit, with positive values indicating wetter-than-usual conditions, and negative values indicating drier-than-usual conditions. If you would like to learn more about the data and how they were generated, see the Methods in Gibb et al. 2023.

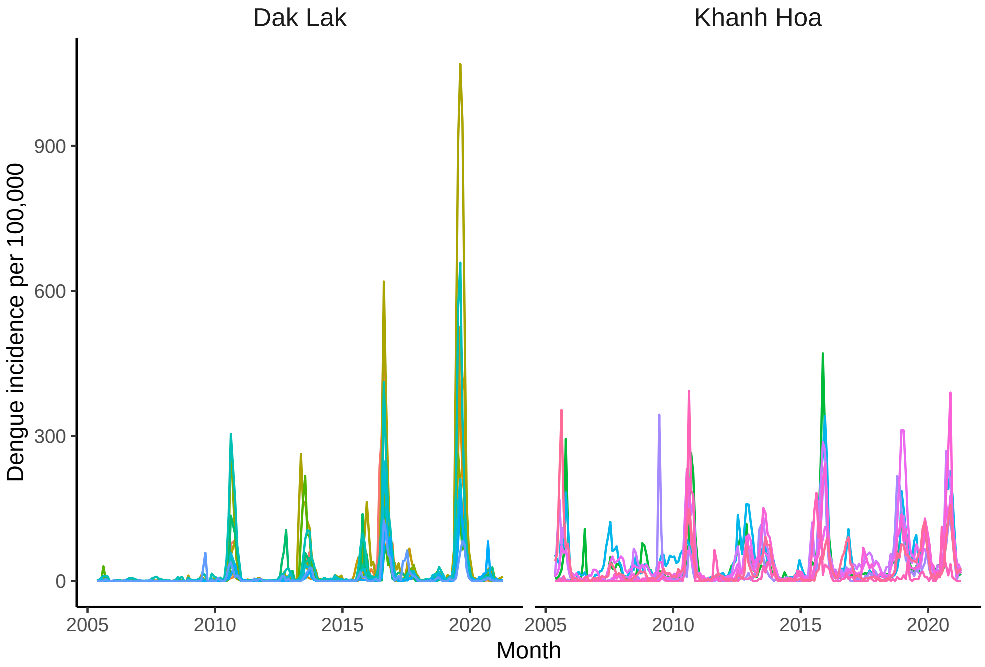
```r
# dengue surveillance data
# format dates
dd = read.csv("./data/dengue/dengue_central_ob.csv") %>%
  dplyr::mutate(date = as.Date(date))

# an accompanying shapefile of vietnam districts
shp = sf::st_read("./data/shapefiles/central.shp")

# shapefile of Vietnam border (for mapping)
shp_vnm = sf::st_read("./data/shapefiles/gadm36_VNM_0.shp")
```

- **Q1**: Explore the dataset. What data do we have and what format are they stored in? How many districts do we have data for, and how many years? Plot a histogram of our outcome variable (dengue cases) - what does this distribution look like?

Let's start by visualising the time series of dengue incidence to understand what historical trends have looked like. Note that the population of each district is different, and also has changed over time, so for visualisation and analysis it is important to adjust our counts to make them comparable between places and times. We can do this by calculating the **dengue incidence rate**, i.e. the number of cases per 100,000 persons.

- How would you describe the dengue epidemic characteristics across these two provinces? Is the incidence similar between years, or variable?

```r
# calculate incidence per 100,000 persons
dd$incidence = dd$cases / (dd$population_census / 100000)

# visualise a histogram of incidence
# what do you notice about this distribution?
ggplot() +
  geom_histogram(data=dd, aes(x=incidence), binwidth=10, fill="skyblue4", color="black") +
  theme_classic() + xlab("Dengue incidence") + ylab("Count")

# visualise time series of incidence
# each coloured line is a district
# broken into separate panels for each province
ggplot() +
  geom_line(data=dd, aes(date, incidence, group=areaid, color=factor(areaid))) +
  theme_classic() +
  facet_wrap(~province, nrow=1) +
  theme(legend.position="none",
        strip.background = element_blank(),
        strip.text = element_text(size=12)) +
  xlab("Month") + ylab("Dengue incidence per 100,000")
```



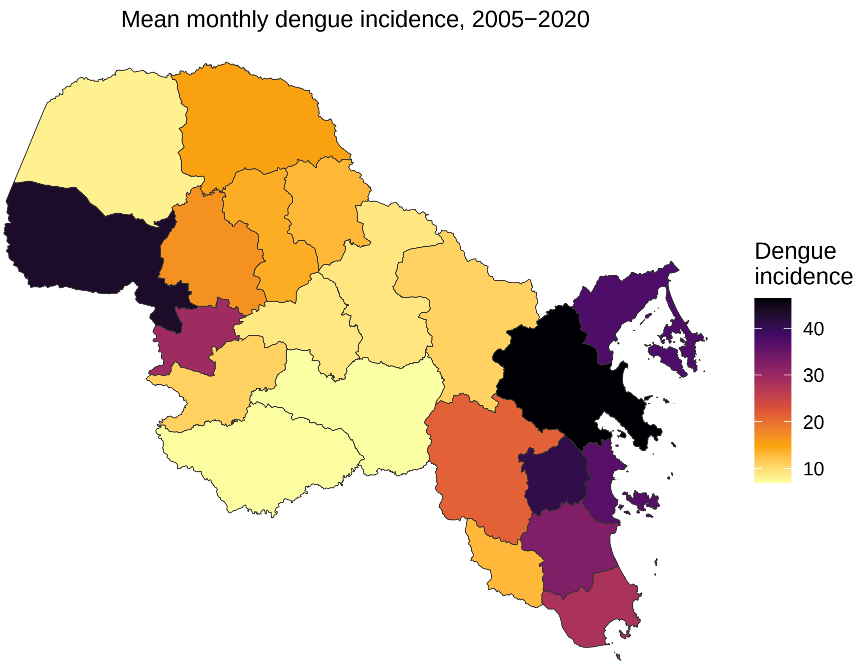
- **Q2**: We are interested in how dengue outbreaks are influenced by the climate. Modify the above code to instead plot the climate indicators (Tmin, precipitation and SPEI-6). How would you describe the climate dynamics in these different areas? How variable is the climate between districts and years?

```
# plot Tmin
ggplot() +
  geom_line(data=dd, aes(date, tmin_1m, group=areaid, color=factor(areaid))) +
  theme_classic() +
  facet_wrap(~province, nrow=1) +
  theme(legend.position="none",
        strip.background = element_blank(),
        strip.text = element_text(size=12)) +
  xlab("Month") + ylab("Average daily minimum temperature (Tmin)")

# plot precipitation
ggplot() +
  geom_line(data=dd, aes(date, precip_1m, group=areaid, color=factor(areaid))) +
  theme_classic() +
  facet_wrap(~province, nrow=1) +
  theme(legend.position="none",
        strip.background = element_blank(),
        strip.text = element_text(size=12)) +
  xlab("Month") + ylab("Precipitation (mm)")

# plot SPEI-6
ggplot() +
  geom_line(data=dd, aes(date, spei6_4m, group=areaid, color=factor(areaid))) +
  theme_classic() +
  facet_wrap(~province, nrow=1) +
  theme(legend.position="none",
        strip.background = element_blank(),
        strip.text = element_text(size=12)) +
  xlab("Month") + ylab("SPEI-6")
```

Finally, we can also map the data to understand how dengue incidence differs between areas. Since we have monthly data, let's calculate the mean monthly incidence across all observations for each district, then visualise this.

- How would you describe the geographic distribution of dengue? Where is the disease burden highest?

```
# calculate means across entire time series
dd_mean = dd %>%
  dplyr::group_by(areaid) %>%
  dplyr::summarise(incidence = mean(incidence))

# combine with shapefile
shp_dd = shp %>%
  dplyr::left_join(dd_mean)

# plot with vietnam country as background
ggplot() +
  geom_sf(data=shp_vnm, color=NA, fill="grey80") +
  geom_sf(data=shp_dd, aes(fill=incidence), color="grey20") +
  theme_void() +
  scale_fill_viridis_c(option="inferno", direction=-1, name="Dengue\nincidence") +
  ggtitle("Mean monthly dengue incidence, 2005-2020") +
  theme(plot.title = element_text(size=11, hjust=0.5))

# plot just the districts
ggplot() +
  geom_sf(data=shp_dd, aes(fill=incidence), color="grey20") +
  theme_void() +
  scale_fill_viridis_c(option="inferno", direction=-1, name="Dengue\nincidence") +
  ggtitle("Mean monthly dengue incidence, 2005-2020") +
  theme(plot.title = element_text(size=11, hjust=0.5))
```



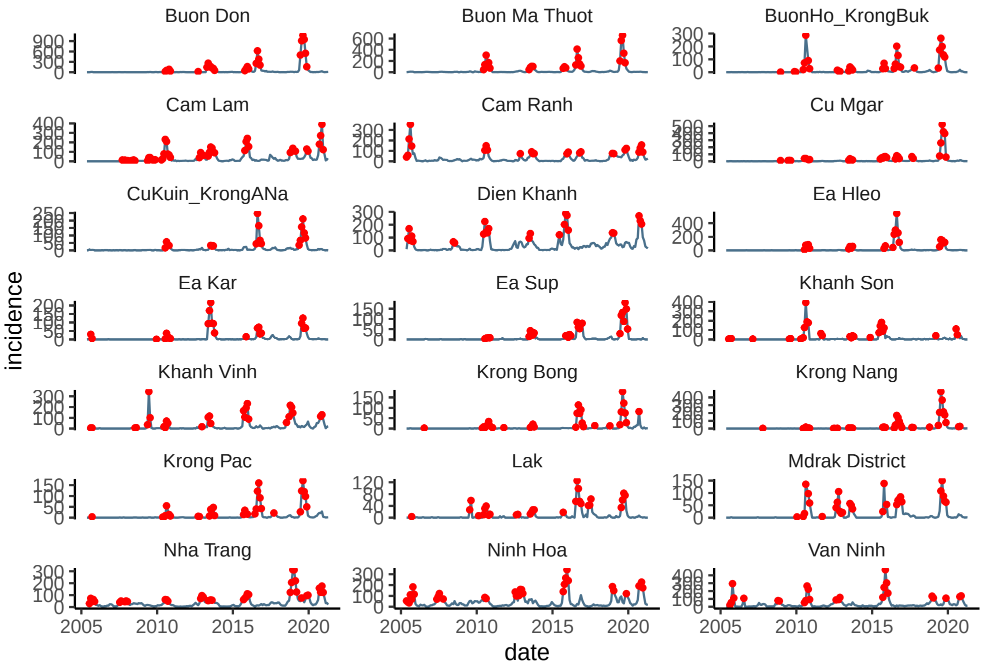
Defining dengue outbreak risk

From visualising the dataset, we can see that dengue epidemic dynamics are highly variable in space and time, leading to very overdispersed data - i.e. lots of observations where the incidence is relatively low, and a few large or extremely large outbreaks, with hundreds of cases. By their nature, predicting the *exact* number of cases during a peak outbreak surge is likely to be very difficult, as we discussed in today's lecture. But this might not matter if we can still do a good job of ***predicting the probability of whether an outbreak will occur at all** (i.e. a binary outcome) - from a health decision making perspective, it is probably still useful to have some advance warning that a very large outbreak is expected to occur.

To do this, we need to have some threshold definition of an "outbreak" - i.e. some threshold number of cases, above which we consider that an "outbreak" is underway. Outbreak thresholds are used a lot for decision-making around endemic diseases, and define an "outbreak" as an anomalously high surge in cases, beyond what would usually be expected for a given time of year. Our data frame contains **two columns with binary outbreak indicators, which are defined in relation to historical patterns of dengue prior to the focal month**. "Outbreak_95" defines an outbreak as when cases exceed the 95th percentile of previous case counts, and "Outbreak_MSD" defines an outbreak as when cases exceed the mean plus 1 standard deviation of previous case counts. **A value of "1" means that an outbreak is considered to be occurring; a value of "0" means that an outbreak is not considered to be occurring**. For more on outbreak definitions, you can check out Brady et al, 2015.

We can visualise historical patterns of dengue and when outbreaks are occurring. For now, we'll use the 95th percentile definition.

```
# plot line graph of incidence
# add points for "outbreak" months
ggplot() +
  geom_line(data=dd, aes(date, incidence), color="skyblue4") +
  geom_point(data=dd[ dd$outbreak_95 ==1, ], aes(date, incidence), color="red", size=1) +
  theme_classic() +
  facet_wrap(~district, scales="free_y", ncol=3) +
  theme(strip.background = element_blank())
```



- **Q3**: Examine and explore the distribution of outbreaks as defined in the "outbreak_95" column. What proportion of months are defined as "outbreak" months? Are there any consistent temporal patterns in outbreaks across months and years? (*Hint: try plotting barplots of the number of outbreaks in each month and year*).
- What information do you think we would need to include in a statistical model to predict dengue outbreak risk?

```
# how many outbreaks? 591 (i.e. 15% of observations)
table(dd$outbreak_95)

# plot number of outbreaks by month
mm = dd %>%
  dplyr::group_by(month_dengue) %>%
  dplyr::summarise(n = sum(outbreak_95))
ggplot() +
  geom_bar(data=mm, aes(month_dengue, n), stat="identity", fill="skyblue4") +
  theme_classic() +
  xlab("Month") + ylab("Num. outbreaks (2005-2020)") +
  scale_x_continuous(breaks=1:12, labels=1:12)

# plot number of outbreaks by year
yy = dd %>%
  dplyr::group_by(year_dengue) %>%
  dplyr::summarise(n = sum(outbreak_95))
ggplot() +
  geom_bar(data=yy, aes(year_dengue, n), stat="identity", fill="skyblue4") +
  theme_classic() +
  xlab("Year") + ylab("Num. outbreaks")
```

Developing a statistical model to predict dengue outbreak risk

To estimate and predict dengue outbreak risk as a function of cliamte and other covariates, we need some kind of **model** to link those covariates to risk. Here, we are modelling outbreaks as a **binomial outcome** (i.e. an outbreak either does or does not occur), so we can do this using logistic regression, where we estimate the contribution of covariates to the log-odds of outbreak occurrence.

The model would be formulated as:

$Y_i \sim Bernoulli(p_i)$

$log(p_i/(1-p_i)) = \beta_0 + X\beta$

where Y_i is the outcome of a Bernoulli trial (either 1 or 0) with probability of success p_i. We use a logit link function to model the log-odds of success as a function of covariates, where β_0 is the intercept, β is a vector of slope parameters, and X is a matrix of covariates.

Just like last week's practical, we will be fitting these models using the *mgcv* package, because this provides a fast and flexible way to fit (generalised) linear **and** nonlinear models in R, and we have good reason to expect that the response of dengue to climate will be nonlinear.

First, we will develop a "baseline" model that does not include any climate information, but *does* include categorical variables for year and month (to account for variable outbreak risks between years and months), and a random intercept for district (to account for overall differences in outbreak risk between different districts). We can use this as a benchmark to evaluate whether adding climate information improves our ability to predict outbreaks.

```r
# set categorical variables as factors
# to ensure model recognises them correctly
dd_m = dd %>%
  dplyr::mutate(areaid = as.factor(areaid),
                year_dengue = as.factor(year_dengue),
                month_dengue = as.factor(month_dengue))

# fit baseline model
# includes random intercept for areaid
# categorical effects of year and month
m_base = mgcv::gam(outbreak_95 ~ s(areaid, bs="re") + year_dengue + month_dengue,
                   data=dd_m,
                   family="binomial",
                   method="REML")
```

- Use *summary* and gratia's *draw* function to examine this model. What does it suggest about how outbreak risk differs between months, years and districts?

```r
# summary of model
summary(m_base)

# we can plot the estimated parameters to examine
# what the model is telling us about the risk over time

# extract fitted parameters
params = as.data.frame(summary(m_base)$p.table)
names(params)[1:2] = c("Estimate", "StdError")
params$param = row.names(params)
params = params %>% dplyr::filter(param != "(Intercept)")
```

```
params$param_type = ifelse(grepl("year", params$param), "year", "month")

# plot estimate and 95% CI for monthly params
# can see the seasonal shape of risk
pm = params %>%
  dplyr::filter(param_type == "month")
pm$month = 2:12
pm %>%
  ggplot() +
  geom_point(aes(month, Estimate)) +
  geom_linerange(aes(month, ymin=Estimate - (1.96*StdError), ymax=Estimate + (1.96*StdError))) +
  theme_classic() +
  ggtitle("Effect of month on dengue risk") +
  xlab("Dengue month") + ylab("Effect on outbreak risk log-odds") +
  theme(plot.title=element_text(size=11, hjust=0.5)) +
  scale_x_continuous(breaks=2:12, labels=2:12)

# plot estimate and 95% CI for year params
# can see lots of variability between years
py = params %>%
  dplyr::filter(param_type == "year")
py$year = 2006:2020
py %>%
  ggplot() +
  geom_point(aes(year, Estimate)) +
  geom_linerange(aes(year, ymin=Estimate - (1.96*StdError), ymax=Estimate + (1.96*StdError))) +
  theme_classic() +
  ggtitle("Effect of year on dengue risk") +
  xlab("Dengue year") + ylab("Effect on outbreak risk log-odds") +
  theme(plot.title=element_text(size=11, hjust=0.5))
```

How can we test this model's predictive ability? We need to set up an experiment, by defining a **training dataset** (which we train the model on) and then a **test dataset** (which the model does not see). We use the trained model to predict outbreaks for each observation in the test dataset, then **statistically assess how successfully the model predicted the outbreaks in the unobserved data**.

For this practical, we will test the model's ability to predict outbreaks during the peak dengue months (months 4 to 9) in the years 2017 to 2020. In this code block, we define our training and test observations, and split the dataset into train and test sets.

```
# create column defining set for each observation
dd$set = ifelse(
  dd$year_dengue %in% 2017:2020 & dd$month_dengue %in% 4:9,
  "test", "train"
)

# split out into train and test datasets
# training on 3,502 obervations
dd_train = dd %>%
  dplyr::filter(set == "train") %>%
  dplyr::mutate(areaid = as.factor(areaid),
                year_dengue = as.factor(year_dengue),
                month_dengue = as.factor(month_dengue))
```

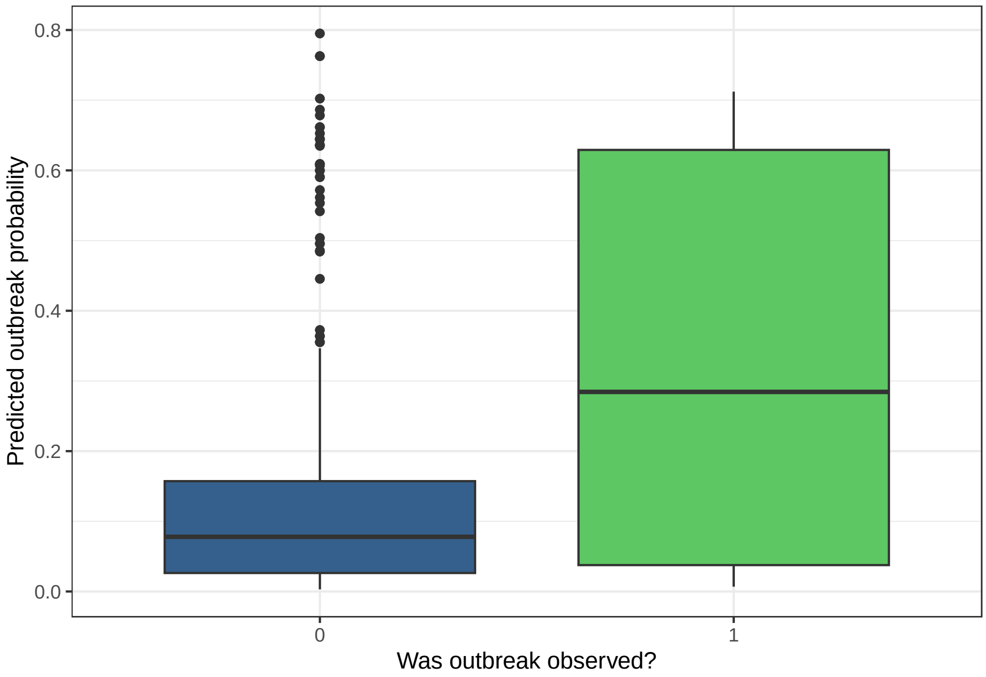
```
# testing on 504 observations (around 13% of data)
dd_test = dd %>%
  dplyr::filter(set == "test") %>%
  dplyr::mutate(areaid = as.factor(areaid),
                year_dengue = as.factor(year_dengue),
                month_dengue = as.factor(month_dengue))
```

- **Q4**: Modify the model fitting code above, to fit a binomial GAM to the training dataset, including a random intercept for areaid, and categorical variables for year and month. **Call this model "m0"**. Examine the model and check you are happy with how it has fitted.

```
# fit model to training data
m0 = mgcv::gam(outbreak_95 ~ s(areaid, bs="re") + year_dengue + month_dengue,
               family="binomial",
               data=dd_train, # specifying "dd_train"
               method="REML")
```

Now we can use the fitted model from Q4 to predict outbreaks in the test dataset. We will do this using the "predict.gam" function, which predicts the probability of an outbreak based on the value of the covariates in the test dataset.

```
# predict for each observation in the test dataset
predicted = mgcv::predict.gam(m0, dd_test, type="response")

# add as a column into the test data
dd_test$predicted = predicted

# plot a boxplot of the predicted probability
# for outbreak and non-outbreak observations
# what do you notice?
dd_test %>%
  ggplot() +
  geom_boxplot(aes(factor(outbreak_95), predicted, fill=factor(outbreak_95))) +
  theme_bw() +
  xlab("Was outbreak observed?") + ylab("Predicted outbreak probability") +
  scale_fill_viridis_d(begin=0.3, end=0.75) +
  theme(legend.position="none")
```



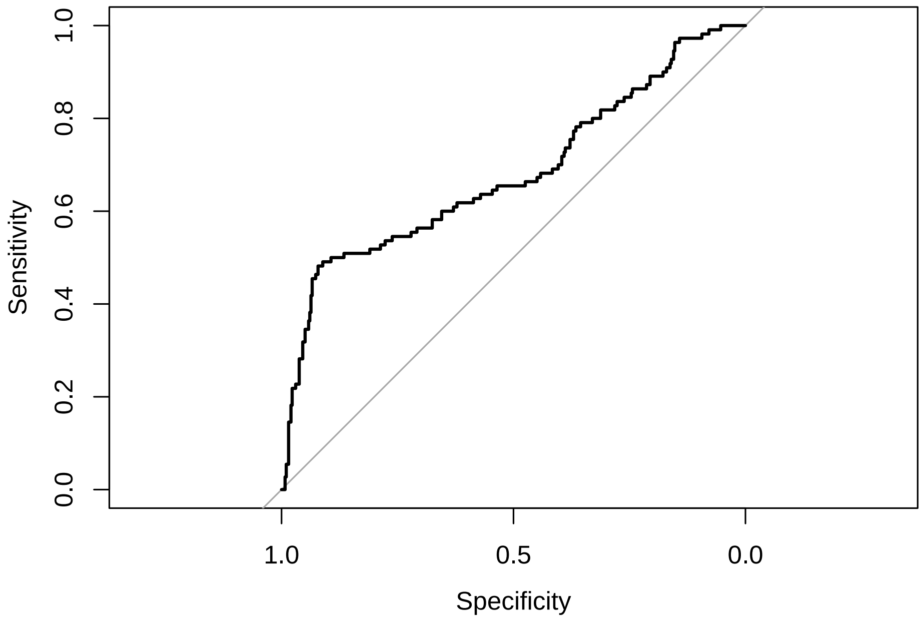
We can see that our model has predicted the *probability of an outbreak* for each observation. However, this is a continuous prediction, and doesn't by itself tell us how good is this model at predicting the binomial outcome, i.e. the occurrence (or not) of an outbreak. To achieve this, we need to apply a cut-off threshold, i.e. a probability threshold above which the model predicts an "outbreak" to be occurring. **But how can we choose the most appropriate cut-off threshold, and how can we determine how well the model predicts outbreaks across a range of possible thresholds?**

Recall from the lecture that predicting a binary outcome involves a trade-off between **sensitivity (proportion of true positives successfully predicted)** against **specificity (proportion of true negatives successfully predicted)**. If we set our cut-off probability threshold very low, we can maximise sensitivity, because we would predict that almost every observation is an outbreak! However, this would sacrifice specificity - i.e. our model would return far too many false alarms, which is unlikely to be useful to inform decisions. Ideally we want a model to successfully predict most of the true outbreaks, but precisely enough that we minimise false alarms.

To determine the optimal threshold, we can generate a receiver-operator curve (ROC), which plots the sensitivity against the specificity for all possible thresholds. The optimal shape of a ROC is to push right up into the top left of the graph, which would mean that the optimal threshold provides very good sensitivity **and** specificity. The **area under the ROC (AUC)** is a good statistic to compare overall accuracy between models, and we can also use the ROC to identify the best cut-off threshold for this model. Let's do this now.

- Work through this code block and ensure you understand each step. Ask if you need help!

```
# generate a ROC curve using observed and predicted values
# how good does the shape of this curve look?
roc_m0 = pROC::roc(dd_test$outbreak_95, dd_test$predicted)
plot(roc_m0)
```



```
# calculate the AUC statistic = 0.68
# an AUC of 1 is perfect accuracy;
# an AUC of 0.5 is no better than a random guess
# how good is this model?
auc_baseline = roc_m0$auc
auc_baseline

# use the roc to extract the best cut-off threshold
# suggested threshold for defining an outbreak = 0.45
# this provides a sensitivity of 0.48 and a specificity of 0.92
# what does this tell you about the model?
cutoff = coords(roc_m0, "best", best.method="youden", transpose = FALSE)

# apply this threshold to predictions
# to produce a binary predicted outcome
dd_test$predicted_binary = as.numeric(dd_test$predicted > cutoff$threshold)

# tabulate observed and predicted
table(dd_test$outbreak_95, dd_test$predicted_binary)
```

What does your analysis of the ROC curve suggest about how good this model is at predicting outbreaks? Let's visualise the relationship between observed and predicted so we can see what's going on. This code plots a heatmap where the colour shading denotes the observed outcome (outbreak or not), then adds an "X" wherever our model **predicts** an outbreak.

```
pred_plot_bl = ggplot() +
  geom_tile(data=dd_test, aes(date, areaid, fill=factor(outbreak_95))) +
  scale_fill_viridis_d(option="magma", direction=-1, begin=0.5, end=0.9, name="Observed\noutbreak") +
  theme_bw() +
  geom_point(data=dd_test[dd_test$predicted_binary==1,], aes(date, areaid), pch=4, size=2) +
  xlab("Date") +
  ylab("District") +
  ggtitle("Baseline (benchmark) model") +
  theme(plot.title = element_text(size=12, hjust=0.5))
pred_plot_bl
```

- How good is our model at predicting outbreaks? What is it doing?

Evaluating whether climate information improves outbreak predictions

Now we have established our benchmark predictive performance using a model without climate information, we can now add climate information to assess whether this helps to improve predictions. Here, we are adding three covariates, all specified as nonlinear (thin-plate spline) effects: Tmin in the preceding month, precipitation in the preceding month, and SPEI-6 4 months earlier. We have chosen these covariates based on a priori knowledge of the system's sensitivity to climate; to understand more on this, please see Gibb et al. 2023.

```
# fitting a new model with climate information
# note that again we are fitting this to the training dataset
m1 = mgcv::gam(outbreak_95 ~ s(areaid, bs="re") +
                 year_dengue + month_dengue +
                 s(tmin_1m, k=9, bs="tp") +
                 s(precip_1m, k=9, bs="tp") +
                 s(spei6_4m, k=9, bs="tp"),
               family="binomial",
               data=dd_train,
               method="REML")
```

- **Q5**: Use gratia's *draw()* function to examine the shape of the fitted climate effects. What do you think these mean? What is your biological interpretation of what is happening?

```
# visualise
gratia::draw(m1)

# hump shaped relationship with Tmin suggesting a thermal optimum around 23C
# likely due to thermal biology of mosquito-borne transmission
# precipitation slightly increases risk but above a threshold (~12mm) starts to decline
# likely due to flushing of breeding sites in extreme rain
# a negative effect of SPEI6 (drought indicator) -
# risk increases during dry (drought) conditions i.e. SPEI6 < 0
# likely due to water storage during drought periods
```

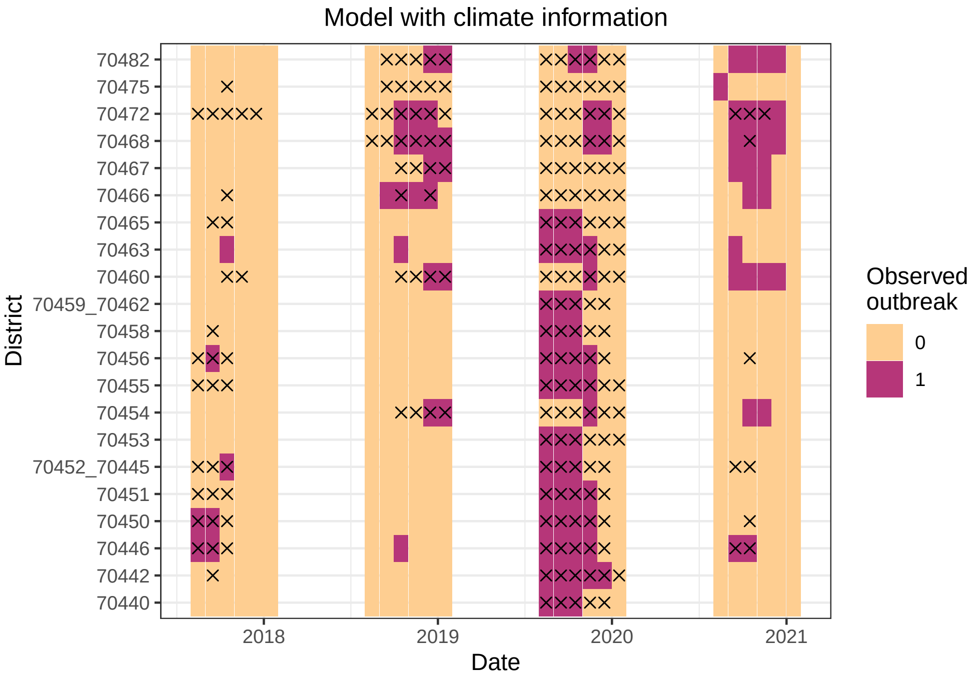
- **Q6**: Modify the code above to use your climate-driven model "m1" to predict outbreak probabilities for the test dataset. Generate and visualise a receiver-operator curve, and use this to calculate an AUC statistic. Has adding climate information improved the model's predictive accuracy?

```
# predict using climate model
predicted = mgcv::predict.gam(m1, dd_test, type="response")
dd_test$predicted = predicted

# generate and plot a ROC - this looks better
roc_m1 = pROC::roc(dd_test$outbreak_95, dd_test$predicted)
```

```
plot(roc_m1)

# calculate AUC = 0.804 (substantially better!)
auc_clim = roc_m1$auc
auc_clim
```

- **Q7**: Modify the code above to identify the optimal cut-off threshold, and examine the specificity and sensitivity of the model at this threshold. Apply this cut-off threshold to predict outbreaks as a binary outcome, then visualise these in relation to observed outbreaks. What does this tell you about **how** climate information has affected this model's predictive accuracy?

```
# calculate threshold
# threshold = 0.08
# specificity 72%, sensitivity 76% - much better balance
cutoff_m1 = coords(roc_m1, "best", best.method="youden", transpose = FALSE)

# apply threshold to predictions
dd_test$predicted_binary = as.numeric(dd_test$predicted > cutoff_m1$threshold)

# plot - what's different this time?
pred_plot_clim = ggplot() +
  geom_tile(data=dd_test, aes(date, areaid, fill=factor(outbreak_95))) +
  scale_fill_viridis_d(option="magma", direction=-1, begin=0.5, end=0.9, name="Observed\noutbreak") +
  theme_bw() +
  geom_point(data=dd_test[dd_test$predicted_binary==1,], aes(date, areaid), pch=4, size=2) +
  xlab("Date") +
  ylab("District") +
  ggtitle("Model with climate information") +
  theme(plot.title = element_text(size=12, hjust=0.5))
pred_plot_clim
```



Predicting outbreaks using a machine learning (random forest) model

So far, we have used a statistical (logistic regression) modelling approach to predict outbreaks based on location, year, month and the climate. There are, however, many alternative modelling methods that we could use to predict binary outbreaks. In the last decade, the use of machine learning tools for classification and prediction has continued to rise within forecasting. One of the most popular methods for this is **random forest**, which makes predictions using an ensemble ("forest") of decision trees built using lots of random subsamples of the dataset. Decision tree-based approaches such as random forest predict outcomes by creating trees that split the dataset at different covariate values, aiming to mimimise predictive error, so unlike statistical models do not have a likelihood function (such as a binomial likelihood) that links observations to model predictions.

We do not have time in this practical (or module!) to dig deeply into the theory and practice of machine learning, but we can explore fitting a random forest model to examine whether it helps to improve our outbreak predictions. We will do this using the R package *ranger* (for fitting random forests) and *caret* (for training and evaluating machine learning models).

When fitting machine learning models, there are numerous options for how the model can be fitted ("parameters") which we need to tune (i.e. select the optimal value for). In the pipeline below, we follow the same approach as earlier, splitting our dataset into train and test sets. We then use the caret package to tune several parameters on our training dataset: **mtry** (number of covariates (features) to split at each node of the decision tree), **splitrule** (the quantitative rule used to determine how a split is chosen at each node of the tree), and **min.node.size** (the smallest number of obervations that can be included in a split). Finally, we use the tuned parameters to fit the best model to the training data, then use that model to predict for the unobserved test data.

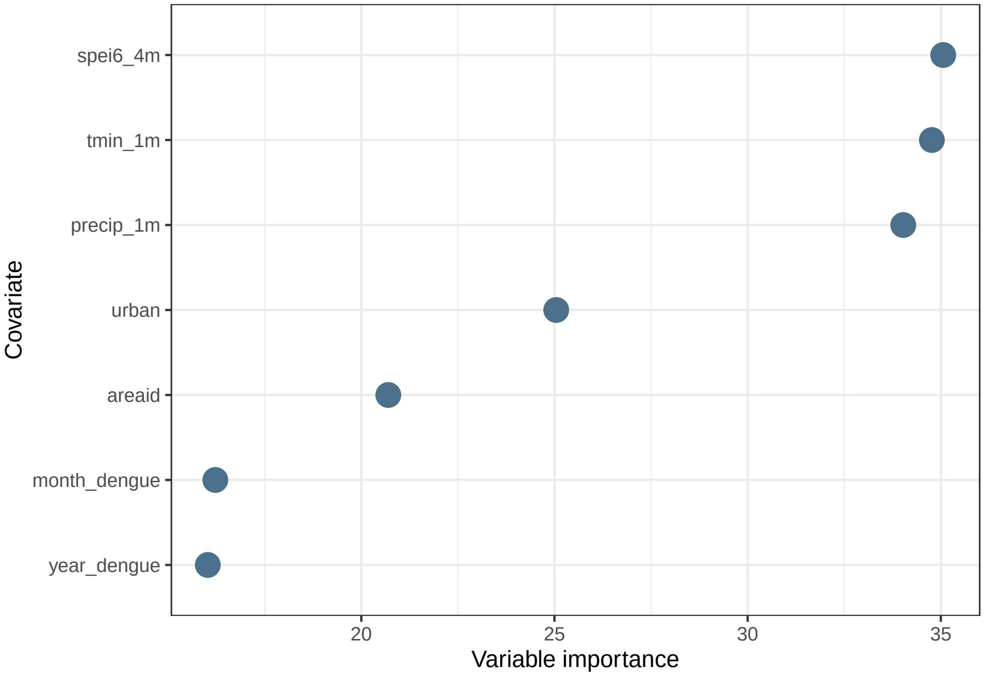
- Work your way through the code block below. Ask if you don't understand anything!

```r
# random forest pipeline

# create training and test datasets as before
dd_train = dd %>%
  dplyr::filter(set == "train") %>%
  dplyr::mutate(areaid = as.factor(areaid),
                year_dengue = as.factor(year_dengue),
                month_dengue = as.factor(month_dengue))
dd_test = dd %>%
  dplyr::filter(set == "test") %>%
  dplyr::mutate(areaid = as.factor(areaid),
                year_dengue = as.factor(year_dengue),
                month_dengue = as.factor(month_dengue))

# specify grid of parameters to tune
# we will try all combinations of these parameters
# and select the best model based on minimising
# predictive error on the "out-of-bag" samples not used
# to train the trees
grid = expand.grid(mtry = c(3,4,5,6),
                   splitrule = c("gini", "extratrees", "hellinger"),
                   min.node.size = c(2, 4, 6, 8, 10))

# specify controls for tuning
# use out-of-bag ("oob") observations for tuning
# run 25 times to allow for variation
fitControl = caret::trainControl(method = "oob", number = 25, verboseIter = FALSE)
```

```
# tune parameters
# our "x" variables are our covariates
# our "y" variable is the outcome
# use 600 trees per model
# n.b. this might take a while
fit = caret::train(
  x = dd_train[ ,  c("areaid", "year_dengue", "month_dengue",
                     "tmin_1m", "precip_1m", "spei6_4m", "urban")],
  y = factor(dd_train$outbreak_95),
  method = 'ranger',
  num.trees = 600,
  tuneGrid = grid,
  trControl = fitControl
  )

# extract the tuned params
tuned = fit$bestTune

# fit model using tuned params
# specify that we want the model to also calculate variable importance
rf_model = ranger::ranger(
  outbreak_95 ~ .,
  data = dd_train[ ,  c("areaid", "year_dengue", "month_dengue",
                        "tmin_1m", "precip_1m", "spei6_4m", "urban", "outbreak_95")],
  num.trees = 600,
  probability = TRUE,
  importance = "impurity",
  mtry = tuned$mtry,
  splitrule = tuned$splitrule,
  min.node.size = tuned$min.node.size
)

# use the fitted model to predict on the test set
test_pred = predict(rf_model, dd_test, fun = function(model, ...) predict(model, ...)$predictions)
dd_test$predicted_rf = test_pred$predictions[ , 2]
hist(dd_test$predicted_rf)
```

We can use the fitted model to examine the **variable importance** - this is a measure of the relative contribution of each covariate to reducing error in the model. Run the next code block and examine the output - what does this suggest about the most important variables helping us to predict outbreaks?

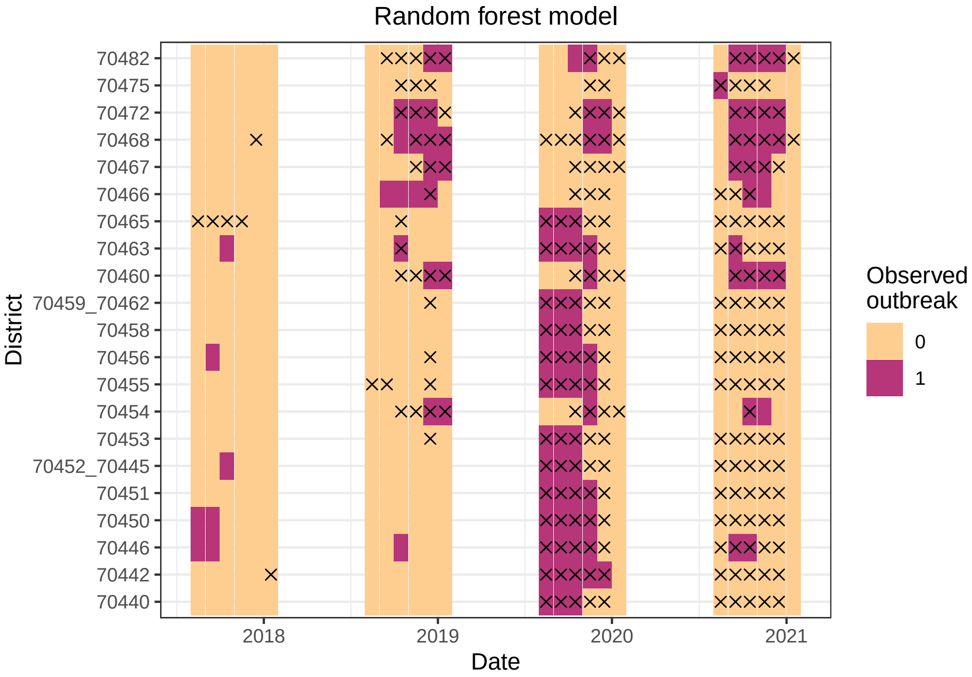
```
# extract variable importance
imp = as.data.frame(ranger::importance(rf_model))

# create a data frame for plotting
imp$param = row.names(imp)
names(imp)[1] = "importance"
imp = imp %>% dplyr::arrange(importance)

# visualise variable importance
imp$param = factor(imp$param, levels=imp$param, ordered=TRUE)
imp %>%
  ggplot() +
  geom_point(aes(param, importance), size=5, color="skyblue4") +
  theme_bw() +
  coord_flip() +
  ylab("Variable importance") +
  xlab("Covariate")
```



- **Q8**: The code block above has generated predictions of outbreak probability from the random forest model. Modify the code from earlier to generate a ROC curve for the random forest model. Use this to calculate the AUC value and identify the optimal cutoff threshold for an outbreak. Apply this threshold and visualise the model predictions. Is the random forest model more accurate than the logistic regression model at predicting outbreaks? What is different about its predictive performance?

```
# generate and plot a ROC
# comparing true outbreaks "outbreak_95"
# to random forest predictions "predicted_rf"
roc_rf = pROC::roc(dd_test$outbreak_95, dd_test$predicted_rf)
plot(roc_rf)

# calculate AUC (very similar)
auc_rf = roc_rf$auc
auc_rf

# calculate threshold
# higher sensitivity but lower specificity than the GAM model
cutoff_rf = coords(roc_rf, "best", best.method="youden", transpose = FALSE)

# apply threshold to predictions
dd_test$predicted_binary_rf = as.numeric(dd_test$predicted_rf > cutoff_rf$threshold)

# plot - what's different this time?
```

```
pred_plot_rf = ggplot() +
  geom_tile(data=dd_test, aes(date, areaid, fill=factor(outbreak_95))) +
  scale_fill_viridis_d(option="magma", direction=-1, begin=0.5, end=0.9, name="Observed\noutbreak") +
  theme_bw() +
  geom_point(data=dd_test[dd_test$predicted_binary_rf==1,], aes(date, areaid), pch=4, size=2) +
  xlab("Date") +
  ylab("District") +
  ggtitle("Random forest model") +
  theme(plot.title = element_text(size=12, hjust=0.5))
pred_plot_rf
```



Extension exercises

The following extension exercises provide an opportunity further expand your analyses of this dataset and practice exploring these modelling methods. **I strongly recommend working through them**, either during the practical session if there is time, or during your own time. The solutions will be uploaded to Moodle next week, with an opportunity to discuss them in class.

Measuring the relative contribution of climate drivers to dengue prediction

During the workshop we ascertained that the logistic regression model including climate information had significantly higher predictive accuracy than the baseline model. However, we did not explore which climatic covariates provided the greatest improvement in predictive accuracy. One way to explore this is to fit models excluding each climate variable at a time, and compare model accuracy.

- **Q9**: Using the same training and test datasets as before, fit three logistic regression models, each excluding one climate covariate at a time from the full model with all climate covariates (i.e. a model with tmin + precip, a model with precip + spei6, etc). Generate a ROC curve for each of these models based on the test dataset. Compare the AUC values across all 3 models, and compare these to the AUC value for the full model with climate covariates. Which covariate provides the most improvement to predictive ability? Why?

```
# create training and test datasets as before
dd_train = dd %>%
  dplyr::filter(set == "train") %>%
  dplyr::mutate(areaid = as.factor(areaid),
                year_dengue = as.factor(year_dengue),
                month_dengue = as.factor(month_dengue))
dd_test = dd %>%
  dplyr::filter(set == "test") %>%
  dplyr::mutate(areaid = as.factor(areaid),
                year_dengue = as.factor(year_dengue),
                month_dengue = as.factor(month_dengue))

# first fit the full model with ALL covariates
m1 = mgcv::gam(outbreak_95 ~ s(areaid, bs="re") +
                 year_dengue +
                 month_dengue +
                 s(tmin_1m, k=9, bs="tp") +
                 s(precip_1m, k=9, bs="tp") +
                 s(spei6_4m, k=9, bs="tp"),
               family="binomial",
               data=dd_train,
               method="REML")
pred_i = predict.gam(m1, dd_test, type="response")
dd_test = dd_test %>% dplyr::mutate(predicted = pred_i)
roc_m1 = pROC::roc(dd_test$outbreak_95, dd_test$predicted)
auc_m1 = roc_m1$auc

# fit three models excluding 1 covariate at a time
# generate ROC and AUC for each

# 1. excluding Tmin
m1a = mgcv::gam(outbreak_95 ~ s(areaid, bs="re") +
                  year_dengue +
                  month_dengue +
                  s(precip_1m, k=9, bs="tp") +
                  s(spei6_4m, k=9, bs="tp"),
                family="binomial",
               data=dd_train, method="REML")
pred_i = predict.gam(m1a, dd_test, type="response")
dd_test = dd_test %>% dplyr::mutate(predicted = pred_i)
roc_m1a = pROC::roc(dd_test$outbreak_95, dd_test$predicted)
auc_tmin = roc_m1a$auc

# 2. excluding precip
m1b = mgcv::gam(outbreak_95 ~ s(areaid, bs="re") +
                  year_dengue +
                  month_dengue +
                  s(tmin_1m, k=9, bs="tp") +
                  s(spei6_4m, k=9, bs="tp"),
                family="binomial",
                data=dd_train,
                method="REML")
pred_i = predict.gam(m1b, dd_test, type="response")
dd_test = dd_test %>% dplyr::mutate(predicted = pred_i)
roc_m1b = pROC::roc(dd_test$outbreak_95, dd_test$predicted)
auc_precip = roc_m1b$auc

# 3. excluding spei
m1c = mgcv::gam(outbreak_95 ~ s(areaid, bs="re") +
                  year_dengue +
                  month_dengue +
                  s(tmin_1m, k=9, bs="tp") +
                  s(precip_1m, k=9, bs="tp"),
                family="binomial",
                data=dd_train,
                method="REML")
pred_i = predict.gam(m1c, dd_test, type="response")
dd_test = dd_test %>% dplyr::mutate(predicted = pred_i)
roc_m1c = pROC::roc(dd_test$outbreak_95, dd_test$predicted)
auc_spei = roc_m1c$auc

# data frame to compare
# the factor contributing most to outbreak prediction is drought (SPEI6)
# because excluding it causes the AUC to drop the most
# (from 0.8 to 0.73)
auc_comparison = data.frame(
  model = c("full", "tmin", "precip", "spei6"),
  auc = c(auc_m1, auc_tmin, auc_precip, auc_spei)
)
auc_comparison
```

Predicting outbreaks using a different outbreak threshold

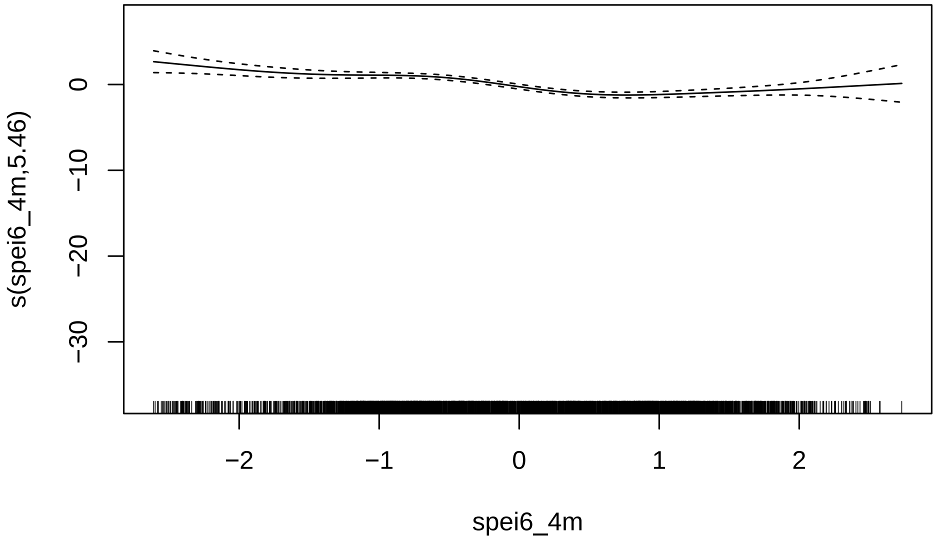
During the workshop we defined an outbreak based on cases exceeding the 95th percentile of historical observations. The data frame contains a second outbreak definition, based on cases exceeding the mean plus 1 standard deviation of historical observations (*"MSD threshold"*). Explore the development of predictive models using this alternative definition, and answer the following questions.

- **Q10**: How many outbreaks occurred in our dataset, according to the mean plus standard deviation ("outbreak_msd") outbreak definition? Is this a more stricter or a more relaxed threshold for an outbreak?

```
# 591 outbreaks for 95th percentile
table(dd$outbreak_95)

# 679 outbreaks for mean+SD
# a less strict threshold for an outbreak
table(dd$outbreak_msd)
```

- **Q11**: Develop a climate-driven model to predict outbreaks using the MSD definition, using the same covariates and the same train-test data split as before. Analyse the ROC curve to assess overall predictive accuracy. Does this suggest it is possible to more accurately (or less accurately) predict outbreaks using the MSD definition compared to the 95th percentile definition?

```r
# create training and test datasets
dd_train = dd %>%
  dplyr::filter(set == "train") %>%
  dplyr::mutate(areaid = as.factor(areaid),
                year_dengue = as.factor(year_dengue),
                month_dengue = as.factor(month_dengue))
dd_test = dd %>%
  dplyr::filter(set == "test") %>%
  dplyr::mutate(areaid = as.factor(areaid),
                year_dengue = as.factor(year_dengue),
                month_dengue = as.factor(month_dengue))

# fit the full climate-driven model
# using "outbreak_msd" as response variable
m1 = mgcv::gam(outbreak_msd ~ s(areaid, bs="re") +
                 year_dengue +
                 month_dengue +
                 s(tmin_1m, k=9, bs="tp") +
                 s(precip_1m, k=9, bs="tp") +
                 s(spei6_4m, k=9, bs="tp"),
               family="binomial",
               data=dd_train,
               method="REML")

# generate roc curve
pred_i = predict.gam(m1, dd_test, type="response")
dd_test = dd_test %>% dplyr::mutate(predicted = pred_i)
roc_m1 = pROC::roc(dd_test$outbreak_msd, dd_test$predicted)
plot(roc_m1)
auc_m1 = roc_m1$auc


# AUC of model for mean + standard deviation threshold is very similar
# similar predictive skill overall
auc_m1

# plot fitted climate functions for MSD model
# very similar to 95th percentile
plot(m1)
```



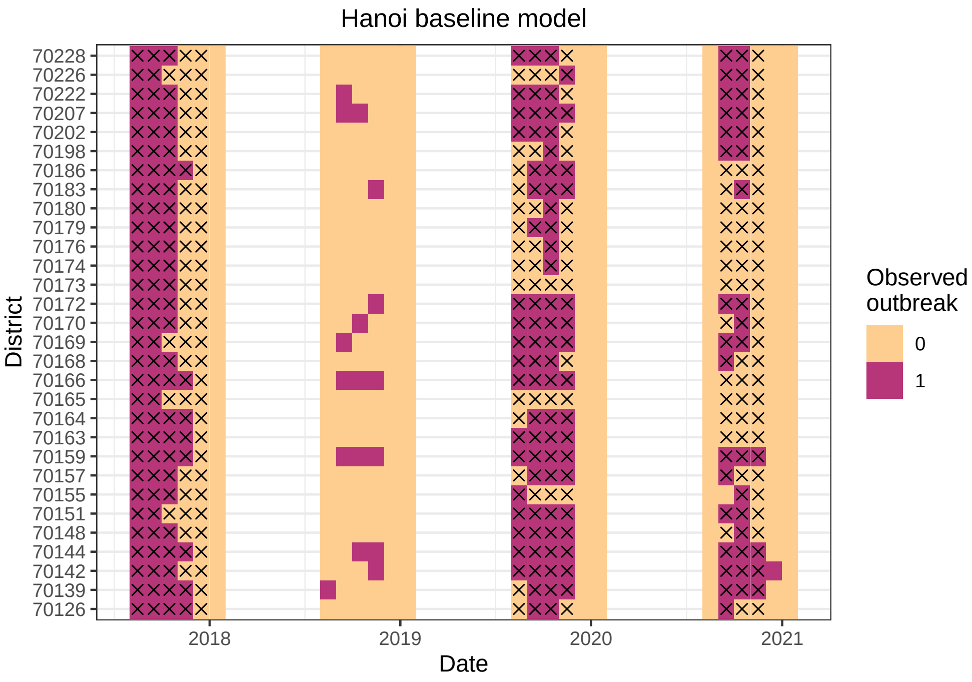
Applying a predictive modelling framework for a different region

The GitHub also contains the same historical dataset for a different part of Vietnam: the northern capital city of Hanoi. Dengue has historically not been a major endemic problem in northern Vietnam because the winter climate is too cool to support sustained year-round transmission. However, dengue is growing as a problem in Hanoi especially, potentially as a result of climate warming. The data folder and shapefiles folder contain an additional set of outbreak and climate data for Hanoi.

- **Q12**: Modify the workshop code to develop a logistic regression model to predict dengue outbreaks in Hanoi, based on the 95th percentile definition. Using the same train-test split as before, develop a baseline model with no climate information, and use a ROC analysis to assess its predictive ability for outbreaks. How accurate is this model?

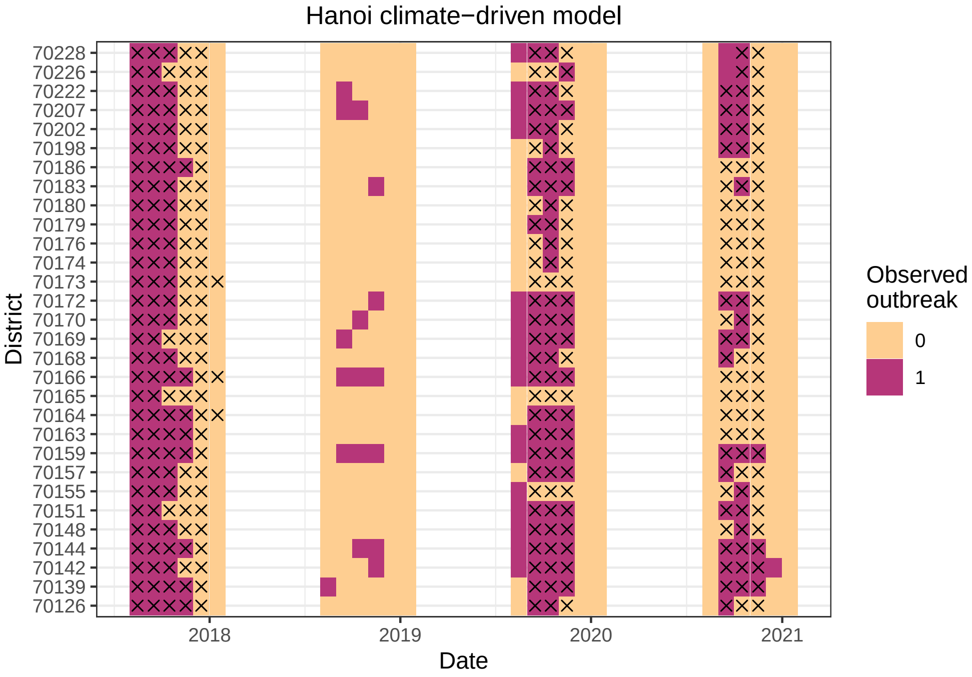
```
# -------- read data -----------

# hanoi surveillance data and shapefile
dd = read.csv("./data/dengue/dengue_hanoi_ob.csv") %>%
  dplyr::mutate(date = as.Date(date))
shp = sf::st_read("./data/shapefiles/hanoi.shp")
shp_vnm = sf::st_read("./data/shapefiles/gadm36_VNM_0.shp")


# ------- initial visualisation ---------

# visualise incidence time series
# a lot of variation between years and some indication that dengue is increasing over time
dd$incidence = dd$cases / (dd$population_census / 100000)
ggplot() +
  geom_line(data=dd, aes(date, incidence, group=areaid, color=factor(areaid))) +
  theme_classic() +
  facet_wrap(~province, nrow=1) +
  theme(legend.position="none",
        strip.background = element_blank(),
        strip.text = element_text(size=12)) +
  xlab("Month") + ylab("Dengue incidence per 100,000")

# graph incidence per district with outbreaks as points
# lots of variation between districts - some with very few cases and low incidence
# also lots of variation in outbreak years between different locations
ggplot() +
  geom_line(data=dd, aes(date, incidence), color="skyblue4") +
  geom_point(data=dd[ dd$outbreak_95 ==1, ], aes(date, incidence), color="red", size=1) +
  theme_classic() +
  facet_wrap(~district, scales="free_y", ncol=5) +
  theme(strip.background = element_blank())

# ----- split into training and test datasets -----

dd$set = ifelse(
  dd$year_dengue %in% 2017:2020 & dd$month_dengue %in% 4:9,
  "test", "train"
)
dd_train = dd %>%
  dplyr::filter(set == "train") %>%
  dplyr::mutate(areaid = as.factor(areaid),
                year_dengue = as.factor(year_dengue),
                month_dengue = as.factor(month_dengue))
dd_test = dd %>%
  dplyr::filter(set == "test") %>%
  dplyr::mutate(areaid = as.factor(areaid),
                year_dengue = as.factor(year_dengue),
                month_dengue = as.factor(month_dengue))

# training on 5,040 observations, testing on 720 observations
nrow(dd_train)
nrow(dd_test)


# ---- fit and evaluate baseline model -----

# fit baseline model
m0 = mgcv::gam(outbreak_95 ~ s(areaid, bs="re") + year_dengue + month_dengue,
               family="binomial",
               data=dd_train, # specifying "dd_train"
               method="REML")

# predict to the test set
pred_i = predict.gam(m0, dd_test, type="response")
dd_test = dd_test %>% dplyr::mutate(predicted = pred_i)

# generate a ROC curve comparing observed to predicted
roc_m0_hanoi = pROC::roc(dd_test$outbreak_95, dd_test$predicted)
plot(roc_m0_hanoi)
```

```r
auc_m0_hanoi = roc_m0_hanoi$auc
auc_m0_hanoi

# the model does pretty well without climate information
# just district, year and month! AIC = 0.82

# examine sensitivity and specificity at the best threshold
# high sensitivity and pretty high specificity
cutoff_m0_hanoi = coords(roc_m0_hanoi, "best", best.method="youden", transpose = FALSE)
cutoff_m0_hanoi

# apply threshold to predictions and plot comparison
dd_test$predicted_binary = as.numeric(dd_test$predicted > cutoff_m0_hanoi$threshold)

# model does a good job overall
# by just predicting 2017, 2019 and 2020 as outbreak years
pred_plot_m0_hanoi = ggplot() +
  geom_tile(data=dd_test, aes(date, areaid, fill=factor(outbreak_95))) +
  scale_fill_viridis_d(option="magma", direction=-1, begin=0.5, end=0.9, name="Observed\noutbreak") +
  theme_bw() +
  geom_point(data=dd_test[dd_test$predicted_binary==1,], aes(date, areaid), pch=4, size=2) +
  xlab("Date") +
  ylab("District") +
  ggtitle("Hanoi baseline model") +
  theme(plot.title = element_text(size=12, hjust=0.5))
pred_plot_m0_hanoi
```



- **Q13**: Compare a climate-driven model for Hanoi, using the same covariates (tmin_1m, precip_1m, spei6_4m), to the baseline model. To what extent does climate information improve predictive accuracy in Hanoi?

```r
# fit climate driven model
m1 = mgcv::gam(outbreak_95 ~ s(areaid, bs="re") +
                 year_dengue +
                 month_dengue +
                 s(tmin_1m, k=9, bs="tp") +
                 s(precip_1m, k=9, bs="tp") +
                 s(spei6_4m, k=9, bs="tp"),
               family="binomial",
               data=dd_train,
               method="REML")

# summary and plots of the model
# much less clear and interpretable climate covariates
# but suggests higher precip and again drought are associated with risk
summary(m1)
plot(m1)

# predict to the test set
pred_i = predict.gam(m1, dd_test, type="response")
dd_test = dd_test %>% dplyr::mutate(predicted = pred_i)

# generate a ROC curve comparing observed to predicted
roc_m1_hanoi = pROC::roc(dd_test$outbreak_95, dd_test$predicted)
plot(roc_m1_hanoi)
auc_m1_hanoi = roc_m1_hanoi$auc
auc_m1_hanoi

# climate information doesn't seem to improve the predictions
# AUC is slightly lower for climate driven model

# at the best threshold, climate info makes the model a bit less sensitive
# and a tiny bit more specific
cutoff_m1_hanoi = coords(roc_m1_hanoi, "best", best.method="youden", transpose = FALSE)
cutoff_m1_hanoi

# apply threshold to predictions and plot comparison
dd_test$predicted_binary = as.numeric(dd_test$predicted > cutoff_m1_hanoi$threshold)

# climate data doesn't provide much additional information
pred_plot_m1_hanoi = ggplot() +
  geom_tile(data=dd_test, aes(date, areaid, fill=factor(outbreak_95))) +
  scale_fill_viridis_d(option="magma", direction=-1, begin=0.5, end=0.9, name="Observed\noutbreak") +
  theme_bw() +
  geom_point(data=dd_test[dd_test$predicted_binary==1,], aes(date, areaid), pch=4, size=2) +
  xlab("Date") +
  ylab("District") +
  ggtitle("Hanoi climate-driven model") +
  theme(plot.title = element_text(size=12, hjust=0.5))
pred_plot_m1_hanoi
```



- **Q14**: Compare the AUC, sensitivity and specificity statistics for the climate-driven Hanoi model, to the same statistics for the climate-driven model you developed for Central Vietnam during the workshop. What does this suggest about our relative ability to forecast outbreaks in Central versus Northern Vietnam? Why might this be the case?

```
# The models for Central and Northern Vietnam had a similar overall AUC of just over 0.8
# However, in Central Vietnam, adding climate information provides a very significant benefit
# and without climate information the AUC is much lower
# In contrast, in Hanoi (North) the baseline model that just uses year and month is more accurate
# and climate information doesn't provide much benefit

# What is going on?

# In Northern Vietnam the dengue transmission setting is very different
# The seasonality is different with hot summers and cool winters
# Transmission only happens during the summer months (rather than year-round in Central regions)
# Importantly, dengue epidemics tend to die out during the winter
# and re-ignite in the summer months, via viruses being reintroduced from further south
# This "emerging" and non-endemic setting means that the outbreak dynamics are more
# unpredictable and sporadic, and different drivers are probably important
# (e.g. human mobility levels), and a suitable climate doesn't mean an outbreak
# will occur

# In the Central regions, transmission occurs year-round and in an endemic cycle
# with dengue viruses persistently circulating
# So when the climatic conditions become more suitable, the likelihood of an outbreak
# is probably higher because the viruses are already there!
# (i.e. don't need to be introduced from anywhere else)

# Take home message:
# Climate information can be very valuable for predicting and forecasting dengue
# However it is much more useful in settings where dengue is endemic
# than in settings where dengue is emerging.
# In emerging settings outbreak dynamics are often inherently more chaotic
# and unpredictable
# So the specific context is very important and the same model
# isn't necessarily useful everywhere!
```