# Hydrological Forecasting in R

Katie Smith
*UK Centre for Ecology & Hydrology*

https://www.ceh.ac.uk/staff/katie-smith

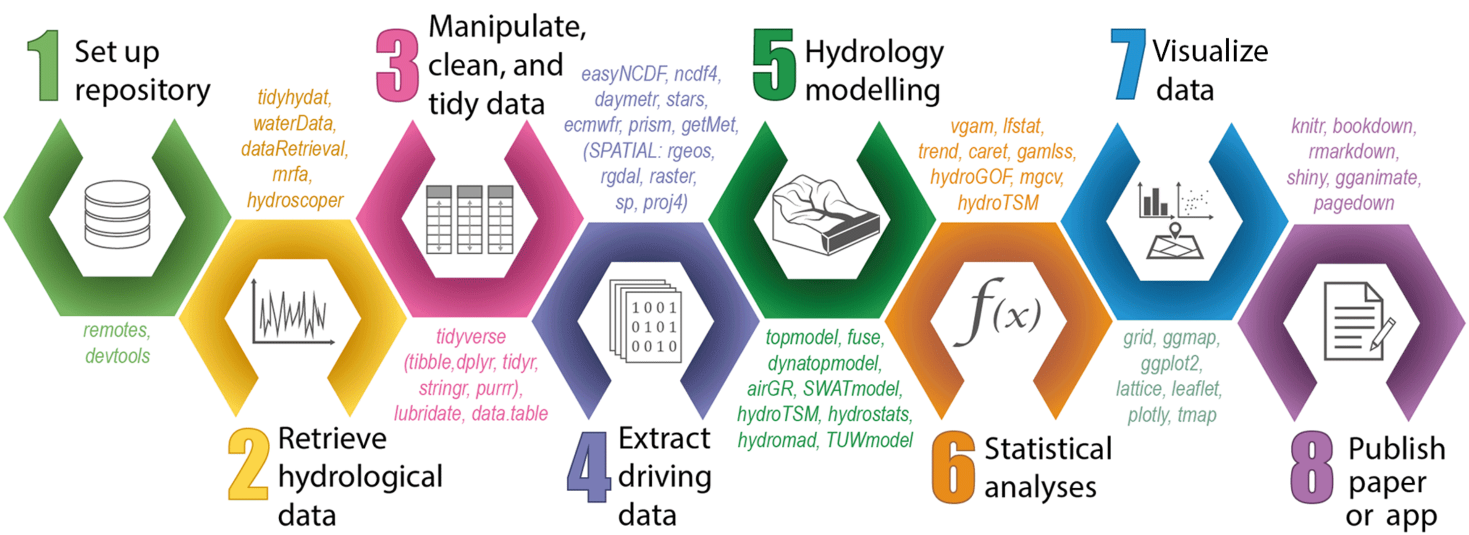
katieasmith26

Using R In Hydrology
vEGU 2021

# Forecasting in R

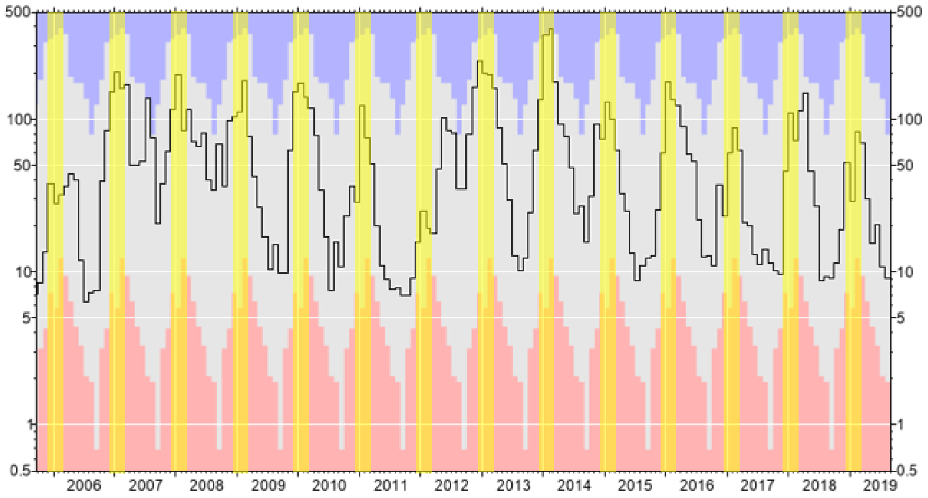
As with any piece of R work, forecasting has a pretty standard workflow of:

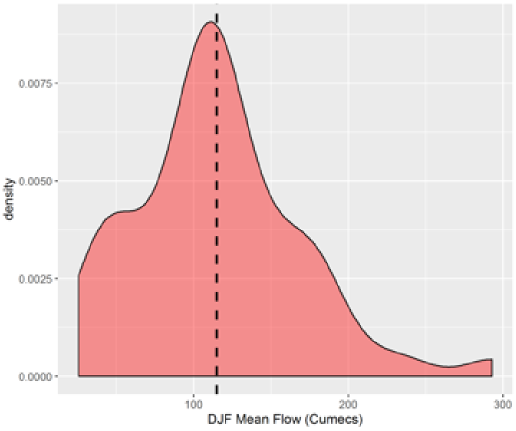
- setting up version control
- retrieving hydrological data for calibration
- tidying data
- extracting driving climate data
- applying a hydrological model
- analysing, visualising, and publishing the results



# Climatology

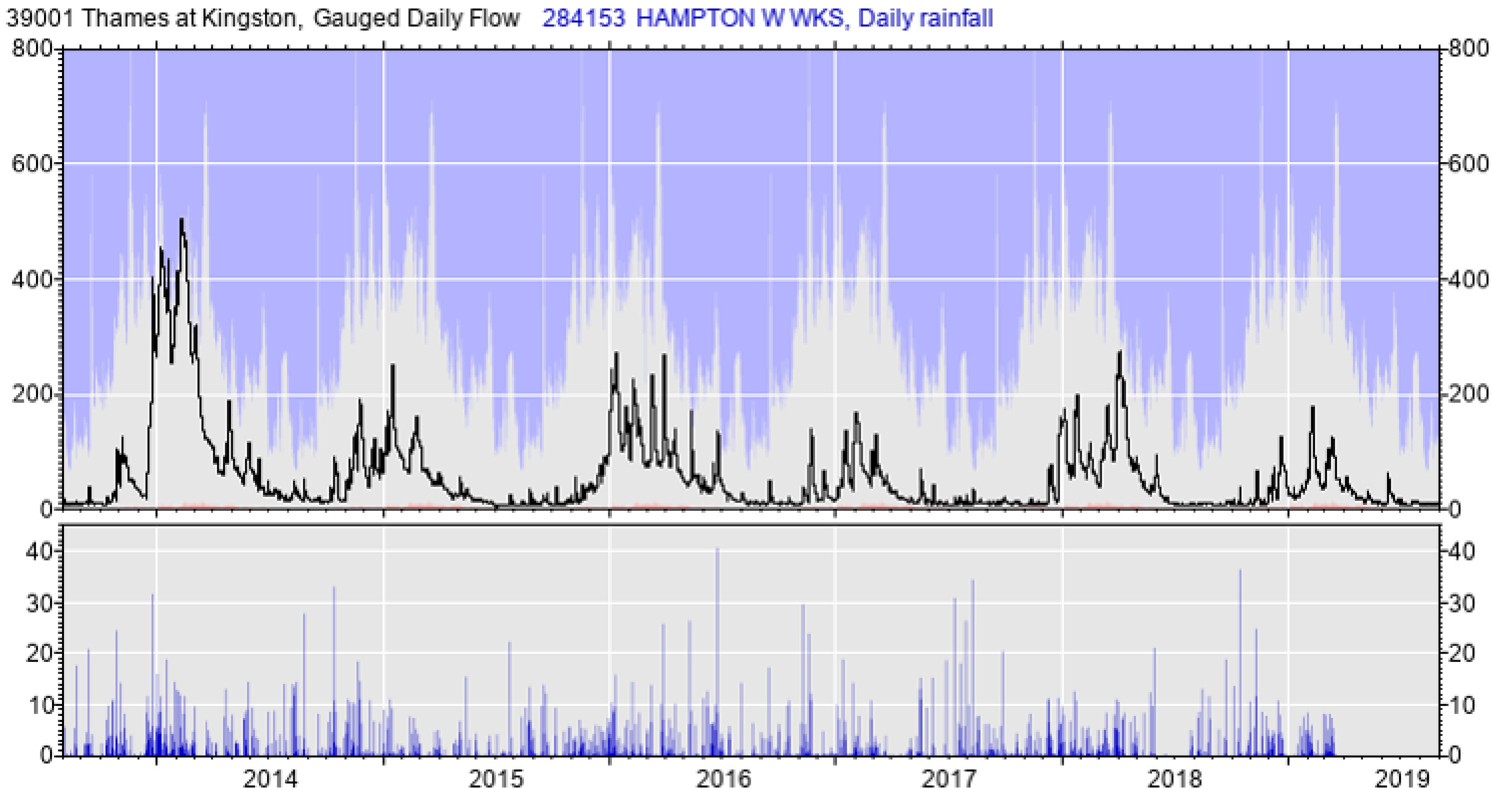
The most basic forecast that we could make is "climatology". This is simply looking at the observed streamflow at this time of year for each year in the past, and using those to form a distribution.



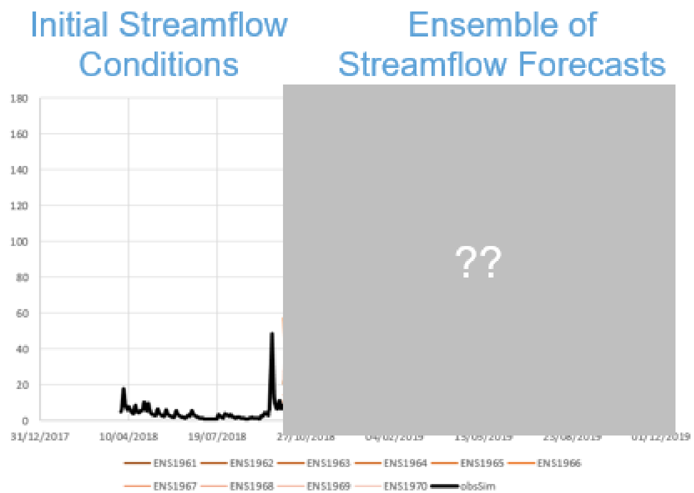
# Ensemble Streamflow Prediction

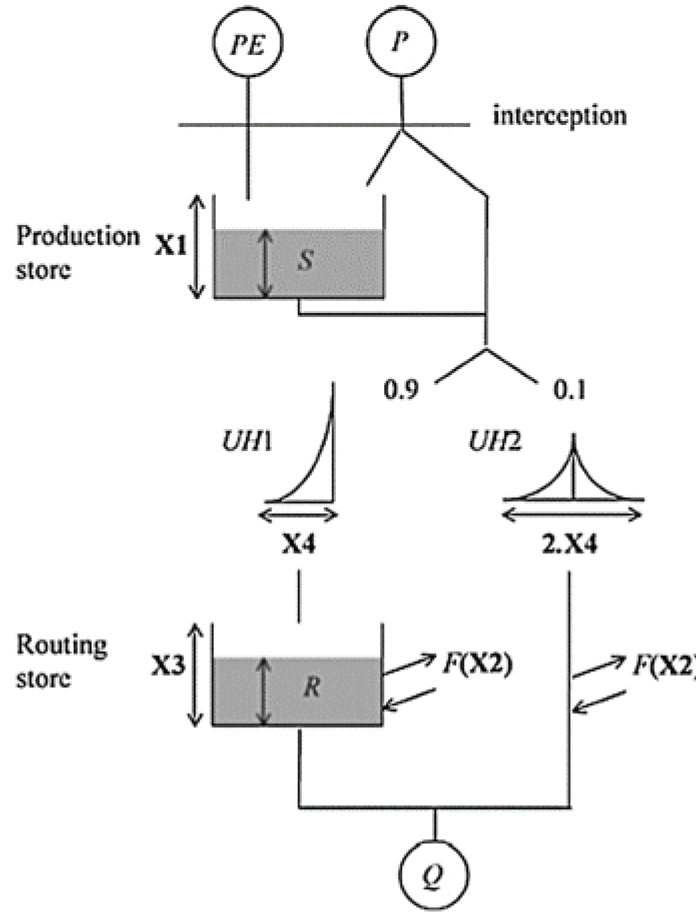
A step up from that is ESP, a very simple method of forecasting that does not require a meteorological forecast. We know our initial conditions, so some of the years in the climatology may not be appropriate.



# Ensemble Streamflow Prediction

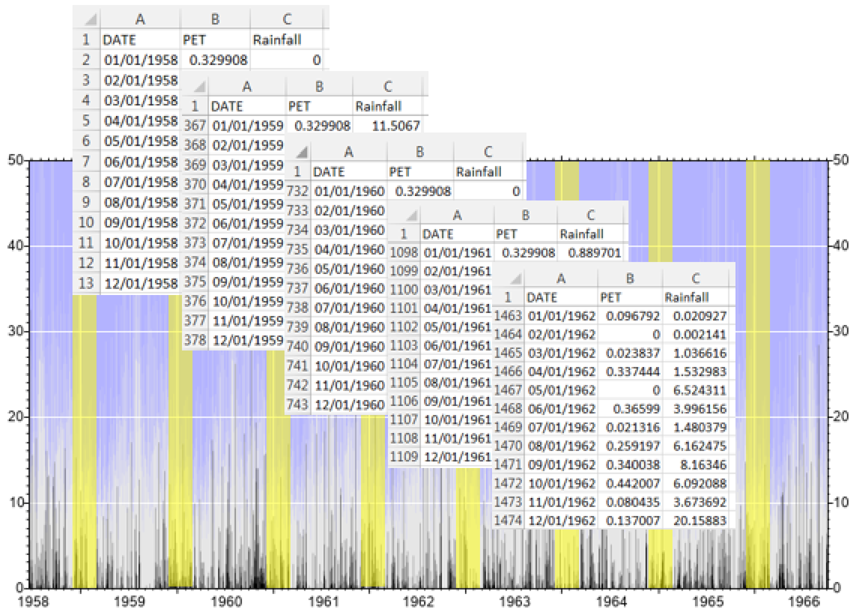
We can model the flow up until the present day, using observed climate data.

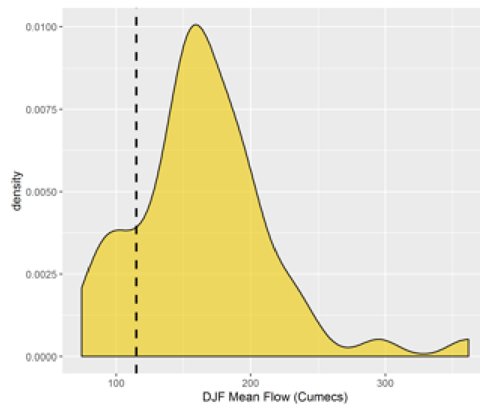


# Ensemble Streamflow Prediction

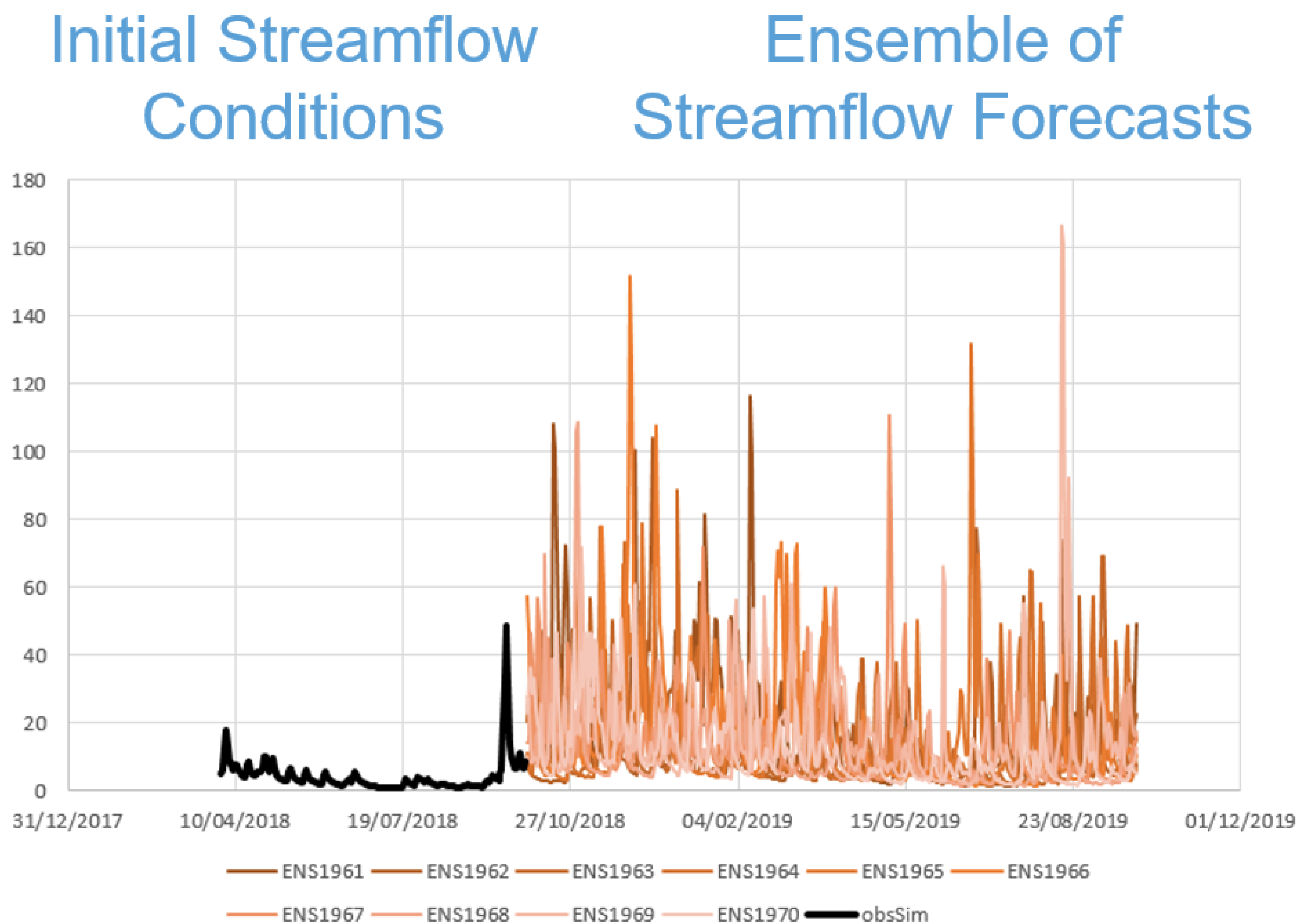
Then ESP simply uses past years of observed climate data, and runs a hydrological model as if that climate were to happen from today.




Dashed line = mean of climatology

# Ensemble Streamflow Prediction



# So let's have a go!

Ensemble Streamflow Prediction with GR4J

The first step is to calibrate the GR4J model. A workthrough on how to do that can be found in the R notebook at:

https://github.com/hydrosoc/rhydro_vEGU21/tree/main/presentations/04.hydro_forecas

# Hydrology in R demonstration airGR - EGU 24/04/2017

Code

Session lead: Katie Smith (Centre for Ecology & Hydrology) k.a.smith@ceh.ac.uk

This is an demonstration of how to use the airGR package of hydrological models in R, as well as how to plot interactive timeseries graphs with the dygraphs package.

## First we need to install some packages

Hide

```
library(xts)
```

```
package <U+393C><U+3E31>xts<U+393C><U+3E32> was built under R version 3.3.3Loading required package: zoo

Attaching package: <U+393C><U+3E31>zoo<U+393C><U+3E32>

The following objects are masked from <U+393C><U+3E31>package:base<U+393C><U+3E32>:

    as.Date, as.Date.numeric
```

Hide

```
library(dygraphs)
```

```
package <U+393C><U+3E31>dygraphs<U+393C><U+3E32> was built under R version 3.3.3
```

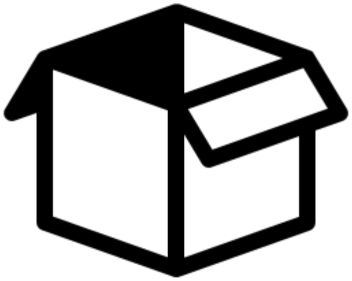
Now we'll load in some observational flow data from the River Thames (naturalised) in England - with thanks to the National River Flow Archive: http://nrfa.ceh.ac.uk/data/search

# You'll need some packages:

```
library(airGR)
library(dplyr)
library(ggplot2)
library(reshape2)
```

I also used a couple of personal functions to make life easier that:

- remove leap years from a date series
- make formatted dates from simple arguments (dd,mm,yyyy)



# Setting Up

We're working on the Lambourn catchment in the UK, a very groundwater dominated catchment. This is where ESP works the best, as it gets a lot of skill from flow persistence.

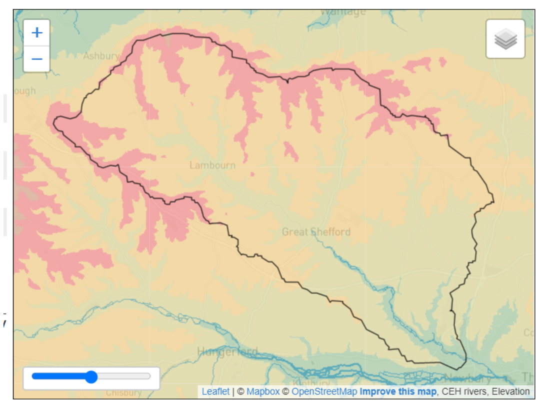
```
# set station and its area
BasinInfoAll<-c(39019,234.1)
getwd()
# Read BasinObs
BasinObs <- read.csv("./Final_New_BasinObs_1961_2017_39019.csv")
BasinObs$DATE <- as.POSIXct(BasinObs$DATE, format = dateFormat)
# Model parameters
Param <- c(455.6977636,-11.6614573,1085.473369,10.2877058)
# set data
DatesR    <- as.POSIXlt(BasinObs$DATE)
Precip    <- BasinObs$PRECIP
PotEvap   <- BasinObs$PET
CatchArea <- BasinInfoAll[2]
```

# Running a baseline

```
# RUN GR4J FOR OBS PERIOD
InputsModel <- CreateInputsModel(FUN_MOD = RunModel_GR4J,
                                 DatesR = DatesR,
                                 Precip = Precip,
                                 PotEvap = PotEvap)
Ind_Run <- seq(1:length(DatesR))
RunOptions <- suppressWarnings(CreateRunOptions(FUN_MOD = RunModel_GR4J,
                                                InputsModel = InputsModel,
                                                IndPeriod_Run = Ind_Run,
                                                IniStates = NULL,
                                                IniResLevels = NULL,
                                                IndPeriod_WarmUp = NULL))
OutputsModel <- RunModel_GR4J(InputsModel = InputsModel,
                              RunOptions  = RunOptions,
                              Param       = Param)
```

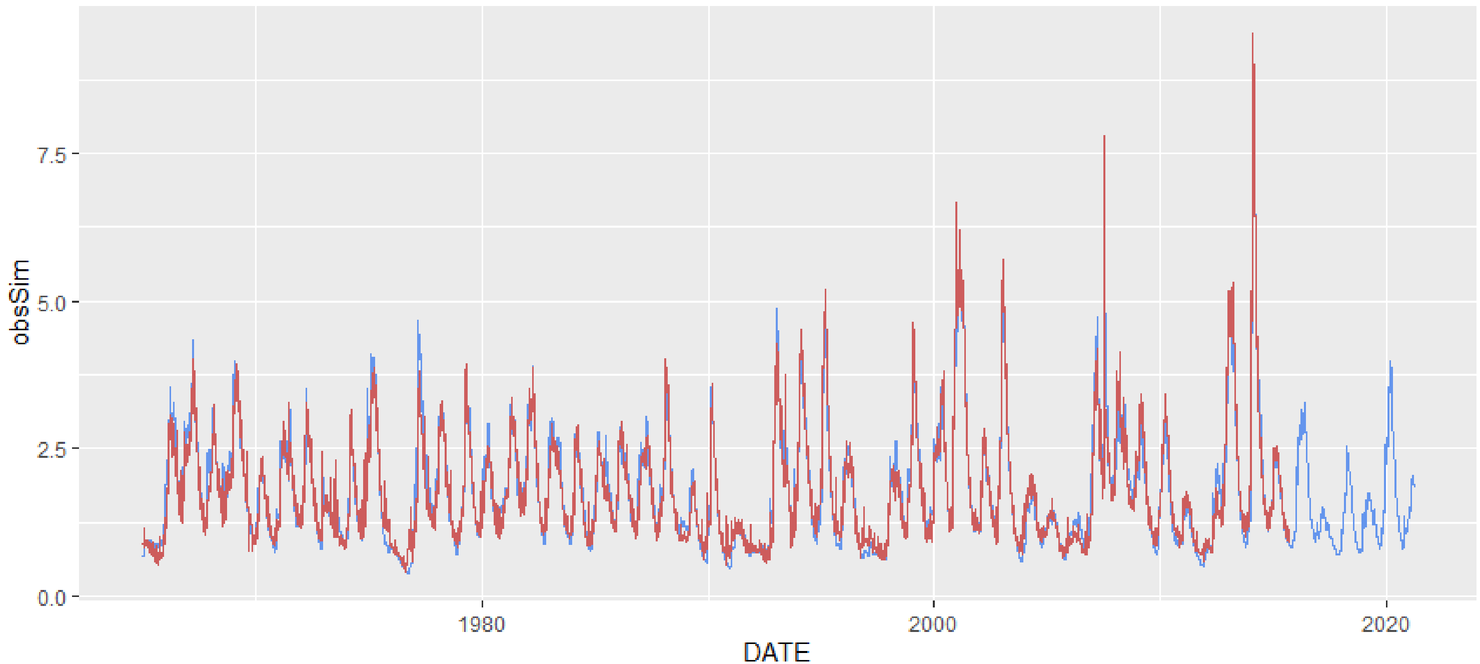
# Post process the output

```
obsSim<-as.data.frame(matrix(ncol=2,nrow=length(BasinObs$DATE),NA))
obsSim[,1]<-BasinObs$DATE
obsSim[,2]<-OutputsModel$Qsim
# Remove first 4 years as warm-up as change DATE to as.Date
colnames(obsSim) <- c("DATE", "obsSim")
obsSim$DATE<-as.Date(obsSim$DATE)
obsSim <- obsSim[which(obsSim$DATE > MakeDate(31,12,1964)), ]
# convert to cumecs
obsSim$obsSim <- (obsSim$obsSim * CatchArea) / 86.4
# merge with obs
obsSim$obs <- BasinObs$FLOW_cumecs[which(BasinObs$DATE==
                                          as.POSIXct(obsSim$DATE[1])):
                                    length(BasinObs$DATE)]
```



# Plotting the baseline run

```
# MAKE A PLOT
ggplot(obsSim)+
  geom_line(aes(x=DATE,y=obsSim),color="cornflower blue")+
  geom_line(aes(x=DATE,y=obs), color="indianred")
```



# Get historic forecast data ready

```r
#remove the first few months to mirror start month of forecast
BasinObs <- read.csv("./Final_New_BasinObs_1961_2017_39019.csv")
BasinObs$DATE <- as.Date(BasinObs$DATE)
BasinObs<-BasinObs[which(BasinObs$DATE >= MakeDate(01,04,1961)),]

#remove leap years from BasinObs
NoLeap <- RemoveLeapDay(BasinObs$DATE)
BasinObsNoLeap <- as.data.frame(NoLeap)
colnames(BasinObsNoLeap)<-c("DATE")
BasinObsNoLeap <- left_join(BasinObsNoLeap, BasinObs, by = "DATE")
```

# Get historic forecast data ready

Make input matrix for Precip

```
PrecipESPin<-as.data.frame(matrix(NA,nrow=length(BasinObsNoLeap$DATE),
                                  ncol=(2021-1961)+1))
colnames(PrecipESPin)<-c("DATE",paste0("ENS",seq(1961,2020)))
PrecipESPin$DATE<-RemoveLeapDay(seq(MakeDate(01,04,1961),
                                    MakeDate(31,03,2021),by="day"))
PrecipESPin[1:length(BasinObsNoLeap$DATE),2:ncol(PrecipESPin)]<-
  BasinObsNoLeap$PRECIP
precipmatrix<-as.data.frame(matrix(BasinObsNoLeap$PRECIP,nrow=365,byrow=F))
colnames(precipmatrix)<-paste0("ENS",seq(1961,2020))
precipmatrix$DATE<-RemoveLeapDay(seq(MakeDate(01,04,2021),
                                     MakeDate(31,03,2022),by="day"))
PrecipESPin<-rbind(PrecipESPin,precipmatrix)
head(PrecipESPin)
```

| | DATE <date> | ENS1961 <dbl> | ENS1962 <dbl> | ENS1963 <dbl> | ENS1964 <dbl> | ENS1965 <dbl> | ENS1966 <dbl> |
|---|---|---|---|---|---|---|---|
| 1 | 1961-04-01 | 0.6130754 | 0.6130754 | 0.6130754 | 0.6130754 | 0.6130754 | 0.6130754 |
| 2 | 1961-04-02 | 3.4946752 | 3.4946752 | 3.4946752 | 3.4946752 | 3.4946752 | 3.4946752 |
| 3 | 1961-04-03 | 8.2962400 | 8.2962400 | 8.2962400 | 8.2962400 | 8.2962400 | 8.2962400 |
| 4 | 1961-04-04 | 8.0246365 | 8.0246365 | 8.0246365 | 8.0246365 | 8.0246365 | 8.0246365 |
| 5 | 1961-04-05 | 1.9650975 | 1.9650975 | 1.9650975 | 1.9650975 | 1.9650975 | 1.9650975 |
| 6 | 1961-04-06 | 0.6230135 | 0.6230135 | 0.6230135 | 0.6230135 | 0.6230135 | 0.6230135 |

6 rows | 1-8 of 61 columns

# Get historic forecast data ready

Do the same for PET

```
PetESPin<-as.data.frame(matrix(NA,nrow=length(BasinObsNoLeap$DATE),
                              ncol=(2021-1961)+1))
colnames(PetESPin)<-c("DATE",paste0("ENS",seq(1961,2020)))
PetESPin$DATE<-RemoveLeapDay(seq(MakeDate(01,04,1961),
                                 (31,03,2021),by="day"))
PetESPin[1:length(BasinObsNoLeap$DATE),2:ncol(PetESPin)]<-
  BasinObsNoLeap$PET
petmatrix<-as.data.frame(matrix(BasinObsNoLeap$PET,
                               nrow=365,byrow=F))
colnames(petmatrix)<-paste0("ENS",seq(1961,2020))
petmatrix$DATE<-RemoveLeapDay(seq(MakeDate(01,04,2021),
                                  MakeDate(31,03,2022),by="day"))
PetESPin<-rbind(PetESPin,petmatrix)
```

And make an output matrix

```
ESPforecasts<-as.data.frame(matrix(NA,nrow=length(BasinObsNoLeap$DATE)+365,
                                  ncol=(2021-1961)+1))
colnames(ESPforecasts)<-c("DATE",paste0("ENS",seq(1961,2020)))
ESPforecasts$DATE<-RemoveLeapDay(seq(MakeDate(01,04,1961),
                                     MakeDate(31,03,2022),by="day"))
```

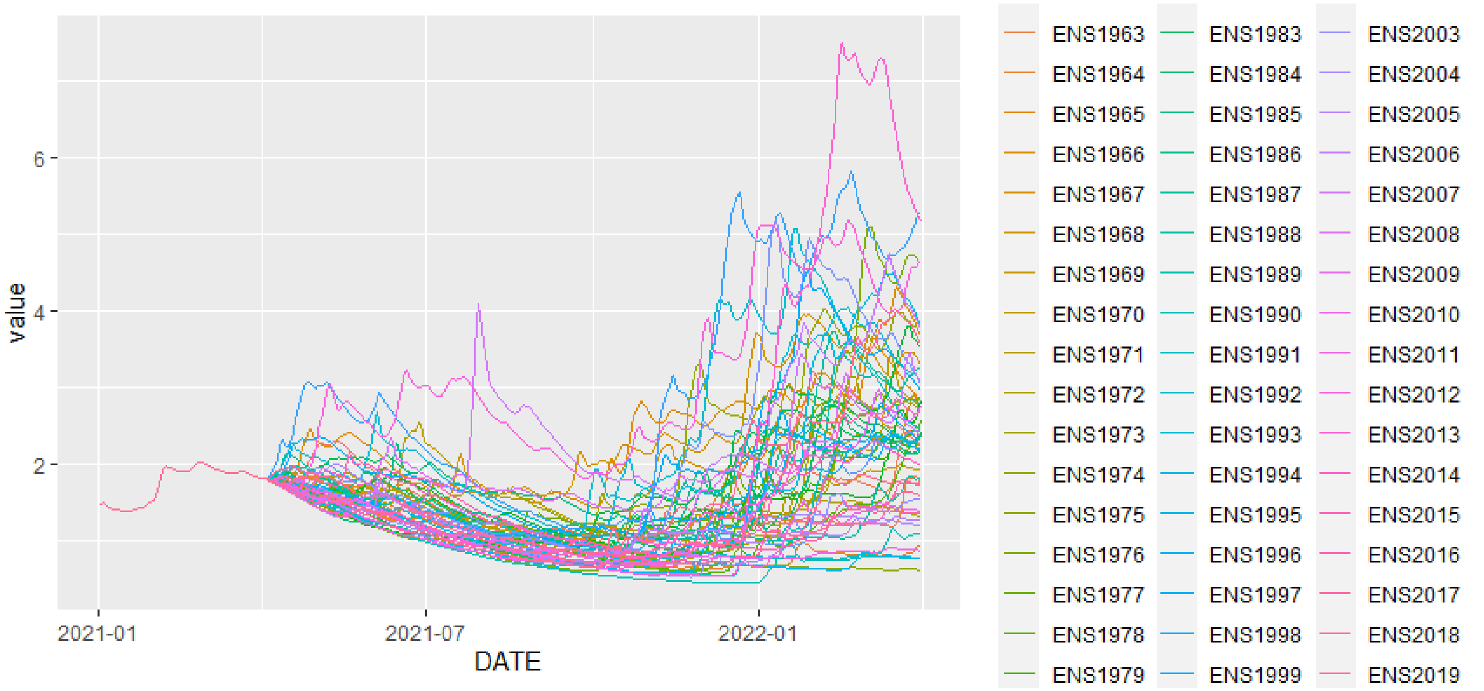
# RUN ESP

```
for (i in 2:61){
  DatesR    <- as.POSIXlt(ESPforecasts$DATE)
  Precip    <- PrecipESPin[,i]
  PotEvap   <- PetESPin[,i]
  CatchArea <- BasinInfoAll[2]
  InputsModel <- CreateInputsModel(FUN_MOD = RunModel_GR4J, DatesR = DatesR,
                                   Precip = Precip, PotEvap = PotEvap)
  Ind_Run <- seq(1:length(DatesR))

  RunOptions <- suppressWarnings(CreateRunOptions(FUN_MOD = RunModel_GR4J,
                                                  InputsModel = InputsModel,
                                                  IndPeriod_Run = Ind_Run,
                                                  IniStates = NULL,
                                                  IniResLevels = NULL,
                                                  IndPeriod_WarmUp = NULL))
    OutputsModel <- RunModel_GR4J(InputsModel = InputsModel,
                                  RunOptions = RunOptions,
                                  Param = Param)
  ESPforecasts[,i]<-OutputsModel$Qsim
}
# convert runoff (mm/day) to flow (M3/s)
ESPforecasts_m3s <- ESPforecasts
ESPforecasts_m3s[,2:61] <- (ESPforecasts[2:61] * CatchArea) / 86.4
```

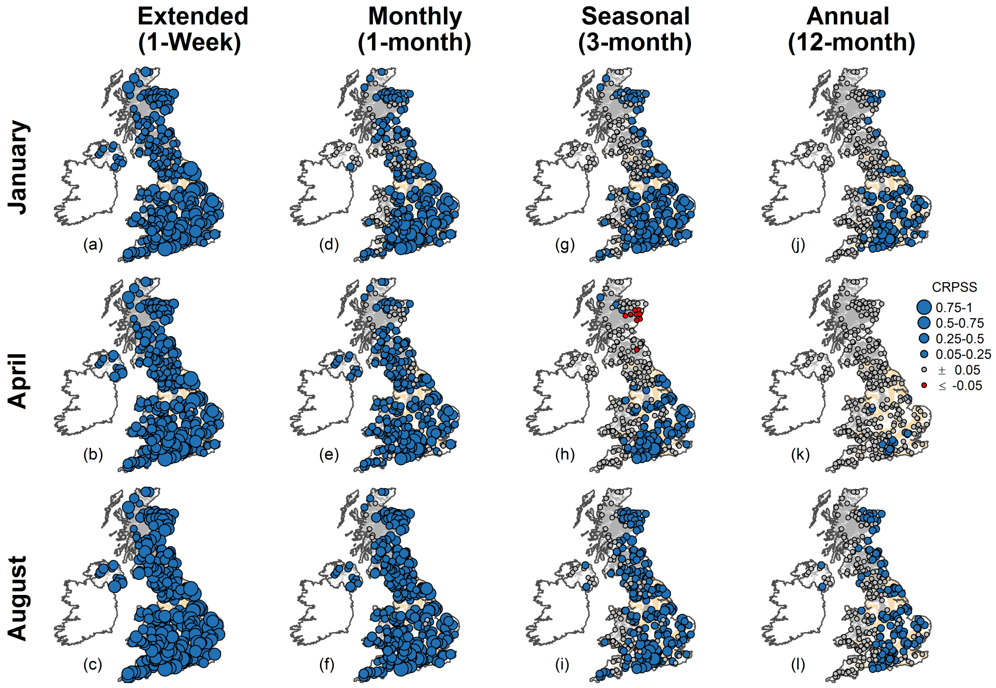
# Plot the forecast

```
# MAKE A PLOT
ESPsub<-ESPforecasts_m3s[which(ESPforecasts$DATE>MakeDate(01,01,2021)),]
ESPmelt <- melt(ESPsub,id.var="DATE")
ggplot(ESPmelt,aes(x=DATE,y=value))+
  geom_line(aes(colour=variable))
```



# Assessing the Skill

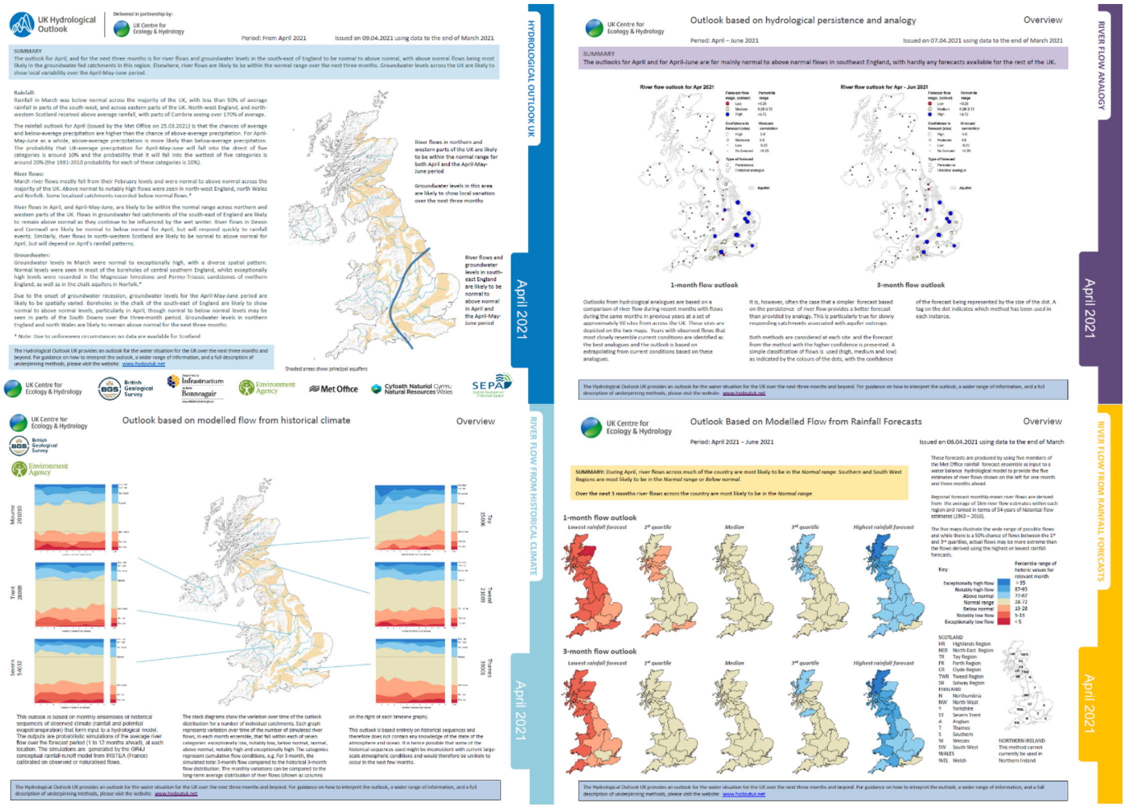
It is important to assess the skill of your forecasting system. You can do this by setting up a "hindcast" experiment (making forecasts of a past date), that you can then compare with observations. For more on this, and how we assessed our system, using the easyVerification R package, see Harrigan et al (2018) https://hess.copernicus.org/articles/22/2023/2018/



# How we use our forecasts

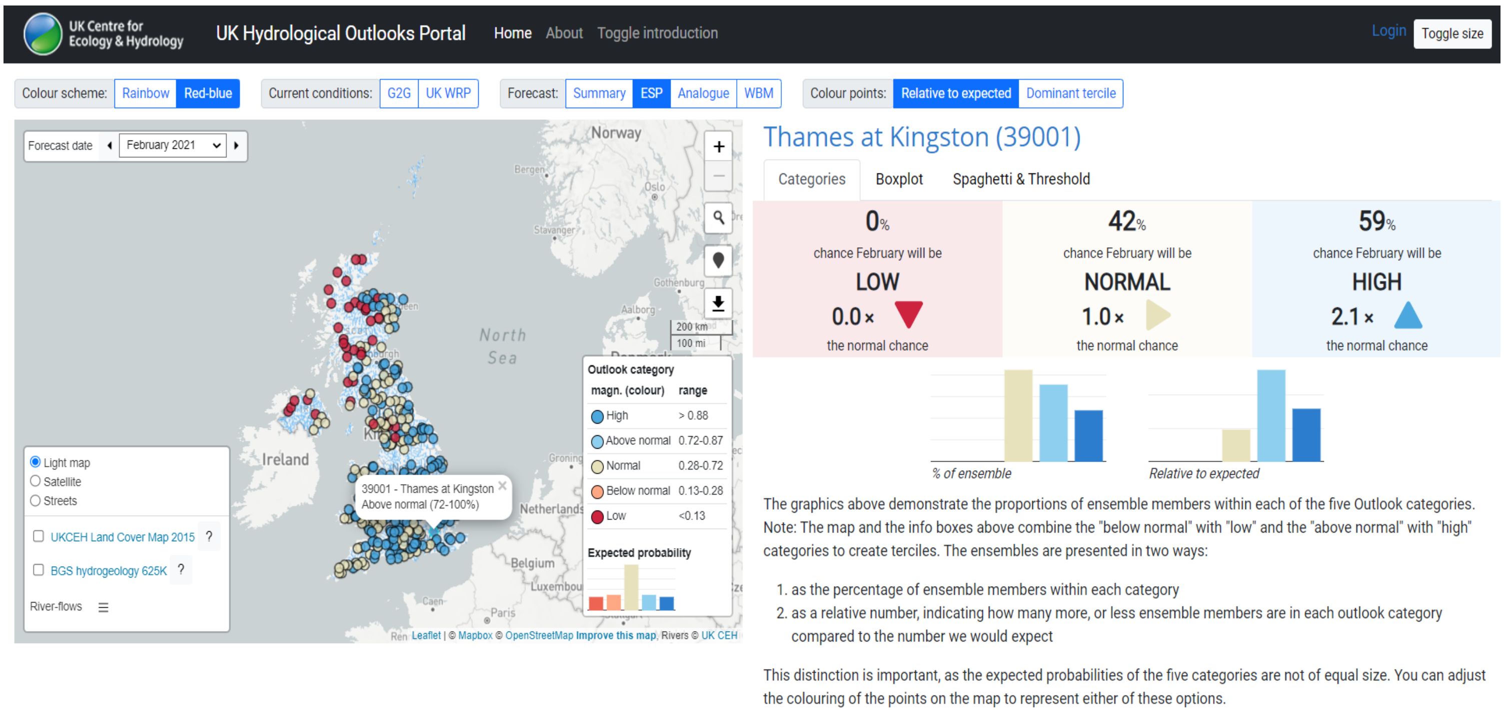
# UK Hydrological Outlook

The ESP method, along with two other methods of hydrological forecasting (1 - persistence and analogues statistical forecasting, and 2 - forcasting using dynamic climate forecasts), are combined with groundwater forecasts to produce our UK Hydrological Outlook.
www.hydoutuk.net



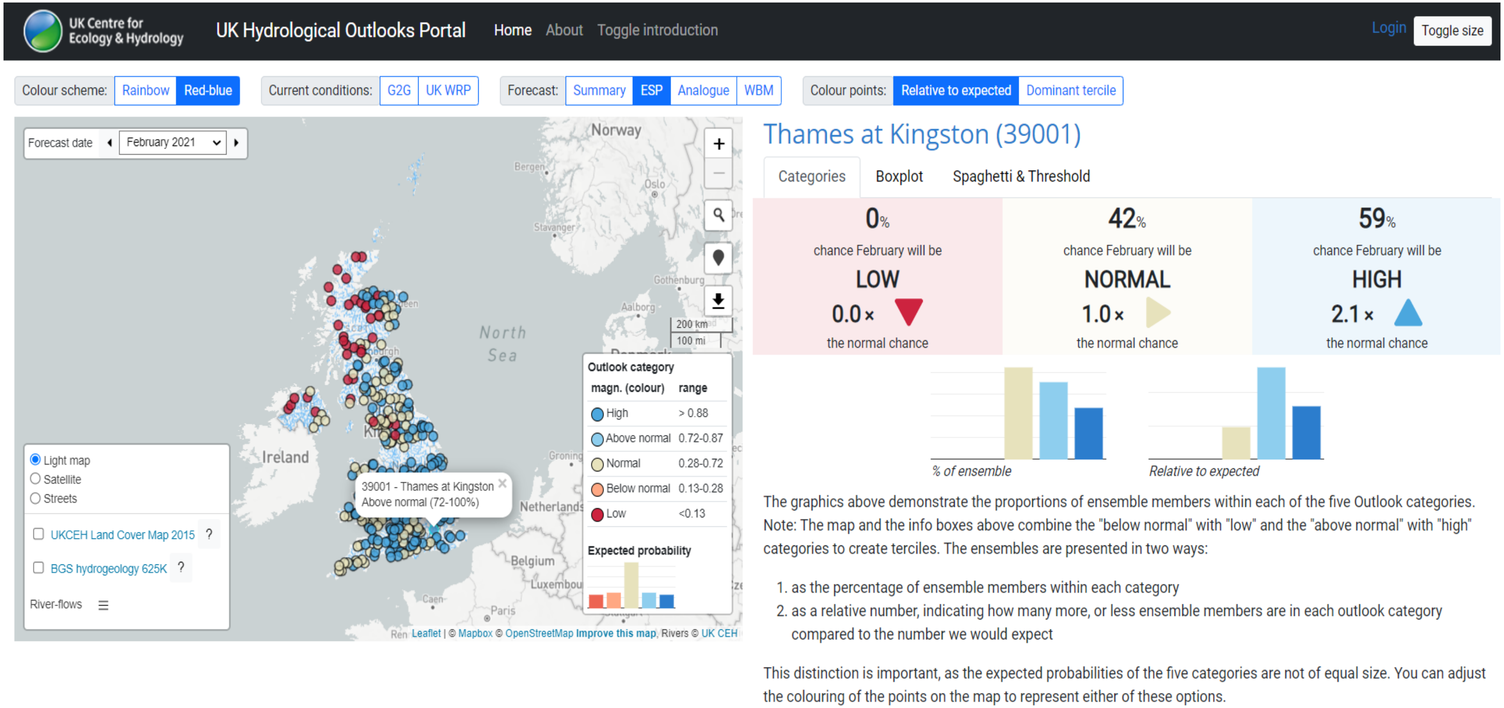
# UK Outlook Portal

We are developing a new interactive web portal, that will allow our users to look at the data in detail. Check back on this link in a couple of months when it will be open to the public.
https://eip.ceh.ac.uk/Hydrology/outlooks/



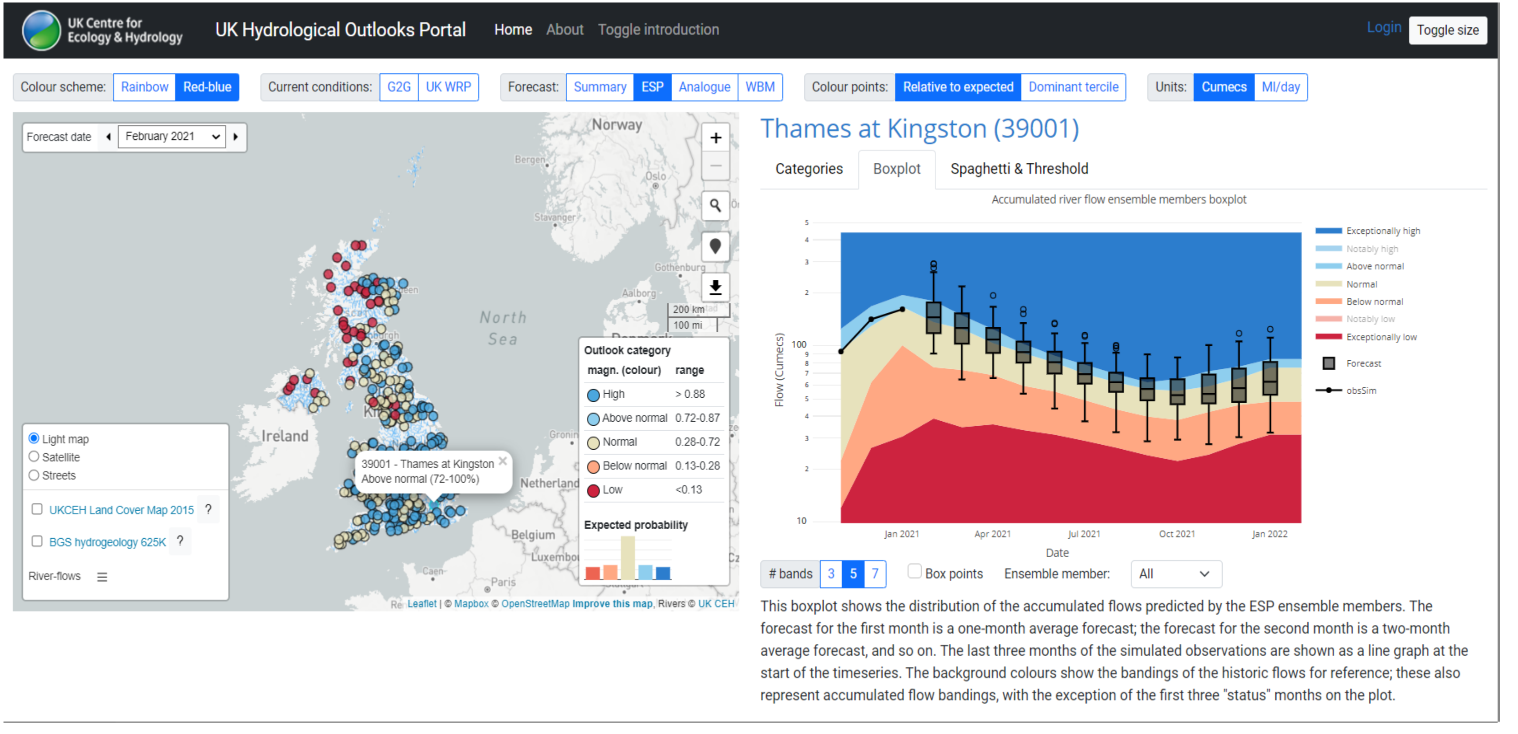
# UK Outlook Portal

This page allows users to look at how many of the ensemble members sit in each category, below normal, normal and above normal. https://eip.ceh.ac.uk/Hydrology/outlooks/



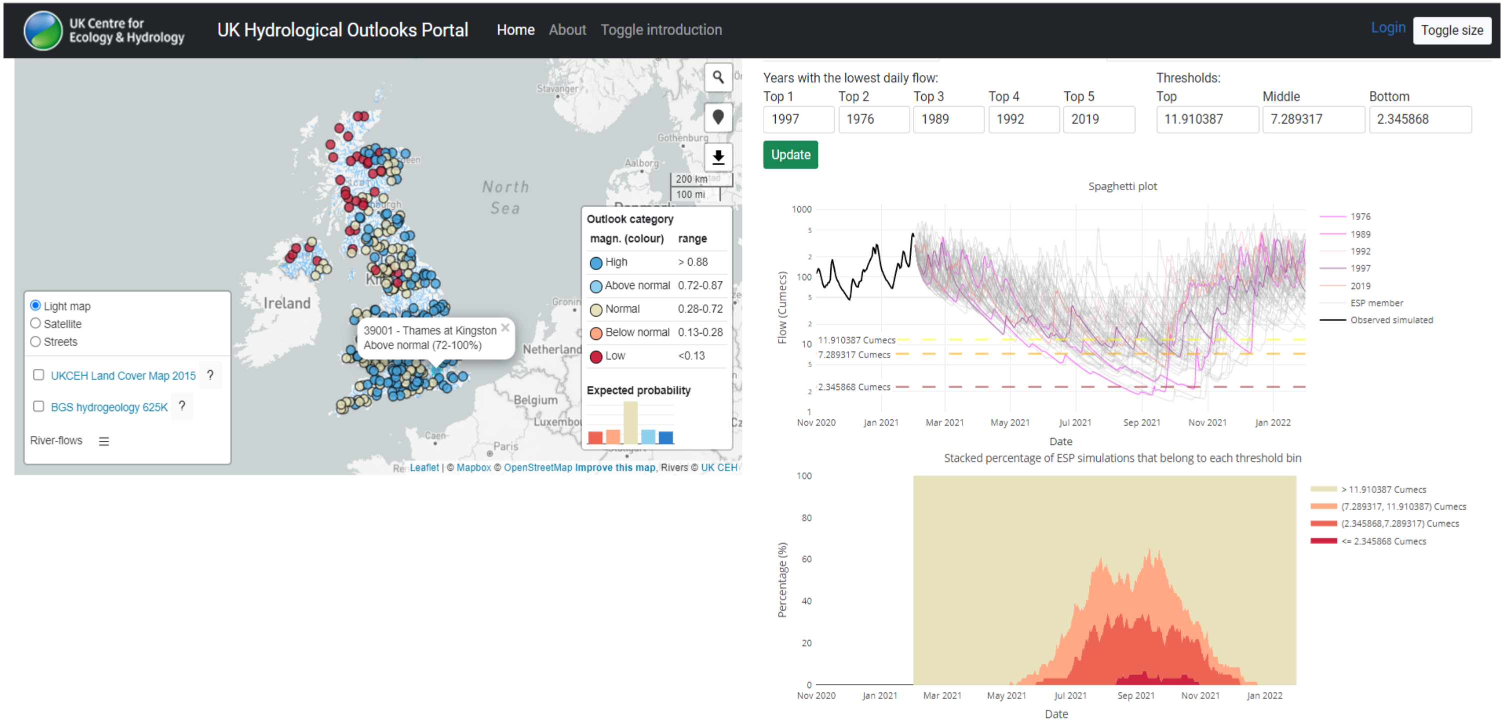
# UK Outlook Portal

This page show users the full ensemble distributions for the forecasts moving forward, against the historic bands of normal flows. https://eip.ceh.ac.uk/Hydrology/outlooks/



# UK Outlook Portal

Finally, this page allows users to select past years of interest to see how the forecast would run if that weather were to happen again now. It also lets users define flow thresholds, and the bottom plot shows the likelihood of flows falling below those thresholds. This is particularly useful for reservoir management. https://eip.ceh.ac.uk/Hydrology/outlooks/



# Thank you for listening!

The slides and materials are available on Github:
https://github.com/hydrosoc/rhydro_vEGU21/