

Kericho

greyhypotheses

The set of external functions used thus far - relative to, therefore based in, GitHub repository [premodelling/time](#) - are

```
sys.source(file = 'R/kericho/StudyData.R', envir = knitr::knit_global())

sys.source(file = 'R/kericho/problems/explore/ExplorationGraph.R',
  envir = knitr::knit_global())
sys.source(file = 'R/kericho/problems/rainfall/Graphs.R',
  envir = knitr::knit_global())
sys.source(file = 'R/kericho/problems/specified/PredictionsGraph.R',
  envir = knitr::knit_global())

sys.source(file = 'R/functions/TimeDependentLag.R', envir = knitr::knit_global())
sys.source(file = 'R/functions/ConfidenceInterval.R', envir = knitr::knit_global())

sys.source(file = 'docs/programme/mathematics/auxiliary_function.R',
  envir = knitr::knit_global())
```

Data Set-up

The original data set, with appended time dependent variables, is

```
'data.frame': 310 obs. of 11 variables:
 $ Year : int 1979 1979 1979 1979 1979 1979 1979 1979 1979 ...
 $ Month : Ord.factor w/ 12 levels "Jan"<"Feb"<"Mar"<...: 1 2 3 4 5 6 7 8 9 10 ...
 $ Cases : int 25 25 20 30 18 18 15 15 10 20 ...
 $ Rain : num 3.7 3.2 5.6 8.3 8.1 5.4 5.5 6.1 5.7 5.6 ...
 $ minT : num 11.8 11.3 10.9 12 10.9 11.4 10.2 10.1 10.2 11.1 ...
 $ maxT : num 24 23.5 25.1 23.6 22.9 22.1 22.1 23 23.9 25.2 ...
 $ VCAP : num 78.5 56.6 131.9 467.6 277 ...
 $ CasesLN: num 3.22 3.22 3 3.4 2.89 ...
 $ datestr: chr "1979-01" "1979-02" "1979-03" "1979-04" ...
 $ date : Date, format: "1979-01-01" "1979-02-01" ...
 $ time : num 0 1 2 3 4 5 6 7 8 9 ...
```

Exploring Relationships

An exploration of the relationship between $\ln(\text{cases})$ and maximum temperature, minimum temperature, and rain.

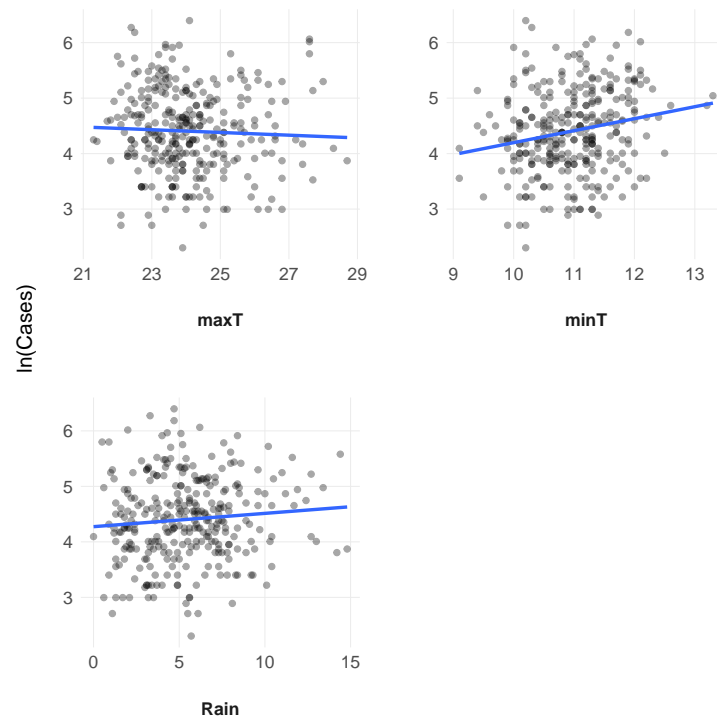


Figure 1: the relationship between $\ln(\text{cases})$ and maximum temperature, minimum temperature, and rain

The Rainfall Series

The function *TimeDependentLag()* creates lagged fields. Hence, expression

```
dataset <- TimeDependentLag(
  frame = instances, frame.date = 'date', frame.date.granularity = 'month',
  variables = 'Rain', lags = seq(from = 0, to = 4) )
```

creates lagged rainfall series; appended to the original data set.

```
# A tibble: 6 x 16
  Year Month Cases Rain minT maxT VCAP CasesLN datestr date time
<int> <ord> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <date> <dbl>
1 1979 Jan     25  3.7 11.8 24   78.5  3.22 1979-01 1979-01-01 0
2 1979 Feb     25  3.2 11.3 23.5 56.6  3.22 1979-02 1979-02-01 1
3 1979 Mar     20  5.6 10.9 25.1 132.  3.00 1979-03 1979-03-01 2
4 1979 Apr     30  8.3 12   23.6 468.  3.40 1979-04 1979-04-01 3
5 1979 May     18  8.1 10.9 22.9 277.  2.89 1979-05 1979-05-01 4
6 1979 Jun     18  5.4 11.4 22.1 132.  2.89 1979-06 1979-06-01 5
# ... with 5 more variables: rain_lag_0 <dbl>, rain_lag_1 <dbl>,
# rain_lag_2 <dbl>, rain_lag_3 <dbl>, rain_lag_4 <dbl>
```

The graphs of *fig. 2* illustrate the relationship between $\ln(\text{cases})$ and each lagged rainfall series. The numeric suffix of each graph's title denotes the rain series lag, in months.

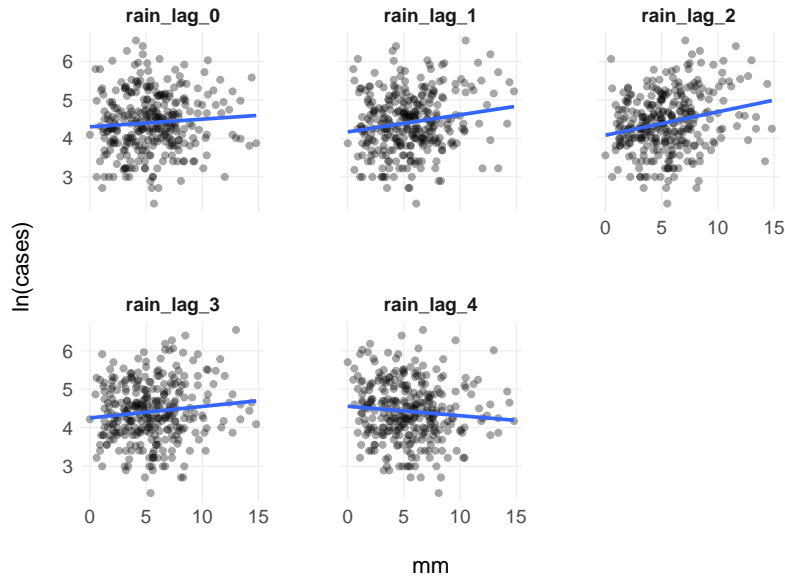


Figure 2: $\ln(\text{cases})$ and the lagged rainfall series. the lags range from 0 to 4 months.

The degree of correlation between $\ln(\text{cases})$ and each lagged rainfall series is quantifiable via the Pearson Correlation Coefficient. The correlation values are:

```
rain_lag_0 rain_lag_1 rain_lag_2 rain_lag_3 rain_lag_4
ln(cases) 0.07433887 0.1662839 0.2280889 0.1124358 -0.09247881
```

Exploring a Specific Model

Considering the time series model

$$\begin{aligned} Y(t) = & \beta_0 + \beta_1 t + \beta_2 I(pmax(t - 50, 0)) + \beta_3 I(t > 225) \\ & + \beta_4 minT(t - k) + \beta_5 maxT(t - k) + \beta_6 Rain(t - k) \\ & + \mathcal{W}(t) + Z(t) \end{aligned} \tag{1}$$

for the Kericho malaria cases data, wherein

variable	description
t	time (months)
$minT$	mininum temperature
$maxT$	maximum temperature
$Rain$	rainfall (millimetres)
k	lag; $k = 2$ months
$\mathcal{W}(t)$	A Matern processwhereby $\kappa = 2.5$
$Z(t)$	Gaussian noise

The function *TimeDependentLag()* creates lagged fields. Hence, the lagged minimum temperature, maximum temperature, and rain fields:

```
variables <- c('minT', 'maxT', 'Rain')

T <- TimeDependentLag(
  frame = instances, frame.date = 'date', frame.date.granularity = 'month',
  variables = variables, lags = seq(from = 2, to = 2))
dataset <- T$frame

'data.frame': 310 obs. of 14 variables:
 $ Year      : int  1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 ...
 $ Month     : Ord.factor w/ 12 levels "Jan"<"Feb"<"Mar"<...: 1 2 3 4 5 6 7 8 9 10 ...
 $ Cases     : int  25 25 20 30 18 18 15 15 10 20 ...
 $ Rain      : num  3.7 3.2 5.6 8.3 8.1 5.4 5.5 6.1 5.7 5.6 ...
 $ minT      : num  11.8 11.3 10.9 12 10.9 11.4 10.2 10.1 10.2 11.1 ...
 $ maxT      : num  24 23.5 25.1 23.6 22.9 22.1 22.1 23 23.9 25.2 ...
 $ VCAP      : num  78.5 56.6 131.9 467.6 277 ...
 $ CasesLN   : num  3.22 3.22 3 3.4 2.89 ...
 $ datestr   : chr  "1979-01" "1979-02" "1979-03" "1979-04" ...
 $ date      : Date, format: "1979-01-01" "1979-02-01" ...
 $ time      : num  0 1 2 3 4 5 6 7 8 9 ...
 $ mint_lag_2: num  NaN NaN 11.8 11.3 10.9 12 10.9 11.4 10.2 10.1 ...
 $ maxt_lag_2: num  NaN NaN 24 23.5 25.1 23.6 22.9 22.1 22.1 23 ...
 $ rain_lag_2: num  NaN NaN 3.7 3.2 5.6 8.3 8.1 5.4 5.5 6.1 ...
```

Exercise 1: Model Fitting

Prior to fitting *Eq. 1*, records that have NaN values ...

```
condition <- !is.na(dataset$rain_lag_2) | !is.na(dataset$mint_lag_2) | !is.na(dataset$maxt_lag_2)
excerpt <- dataset[condition, ]
```

Hence, via the `fit.matern()` function

```
fit2.5 <- fit.matern(
  form = as.formula(log(Cases) ~ time + I(pmax(time - 50, 0)) + I(time > 225)
    + mint_lag_2 + maxt_lag_2 + rain_lag_2),
  time = 'time',
  start.cov.pars = c(1,5),
  kappa = 2.5,
  data = excerpt,
  method = 'nlminb')
```

The summary of the model fitted for *Eq. 1* is

```
Geostatistical linear model
Call:
linear.model.MLE(formula = log(Cases) ~ time + I(pmax(time -
50, 0)) + I(time > 225) + mint_lag_2 + maxt_lag_2 + rain_lag_2,
  coords = as.formula(paste("~", time, "+ t_aux")), data = data,
  kappa = 2.5, start.cov.pars = ..1, method = "nlminb")

              Estimate      StdErr z.value  p.value
(Intercept)    1.5546882    0.5374136   2.8929 0.0038169 **
time           0.0266066    0.0053688   4.9558 7.205e-07 ***
I(pmax(time - 50, 0)) -0.0258646    0.0059857  -4.3211 1.553e-05 ***
I(time > 225)TRUE  0.6931965    0.1792059   3.8682 0.0001097 ***
mint_lag_2      0.1449852    0.0418763   3.4622 0.0005357 ***
maxt_lag_2     -0.0140655    0.0078728  -1.7866 0.0740030 .
rain_lag_2      0.0185512    0.0101402   1.8295 0.0673303 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-likelihood: 98.64848

Covariance parameters Matern function (kappa=2.5)
              Estimate StdErr
log(sigma^2) -1.61326 0.1580
log(phi)     -0.49494 0.1617
log(tau^2)   -2.72287 0.5096

Legend:
sigma^2 = variance of the Gaussian process
phi = scale of the spatial correlation
tau^2 = variance of the nugget effect
```

The parameter σ^2 , ϕ^2 , and τ^2 , and their confidence intervals, are:

```
              estimate lower_ci upper_ci exp(estimate) exp(lower_ci)
log(sigma^2)  -1.613   -1.923   -1.304         0.199         0.146
log(phi)      -0.495   -0.812   -0.178         0.610         0.444
log(tau^2)    -2.723   -3.722   -1.724         0.066         0.024
exp(upper_ci)
log(sigma^2)           0.272
log(phi)              0.837
log(tau^2)            0.178
```

Exercise 2: Predictions

The foci herein are the $\ln(\text{cases})$ point predictions, and their 95% prediction intervals, w.r.t. the months of the Kericho data set. The `time.predict()` function of *auxiliary_function.R*

```
predictor <- time.predict(  
  fitted.model = fit2.5,  
  predictors = excerpt[, c('time', 'mint_lag_2', 'maxt_lag_2', 'rain_lag_2')],  
  time.pred = excerpt$time,  
  scale.pred = 'exponential')
```

creates the `time.predict()` object of predictions, including the confidence intervals. The resulting graph (*fig. 3*) illustrates the curve's confidence intervals band.

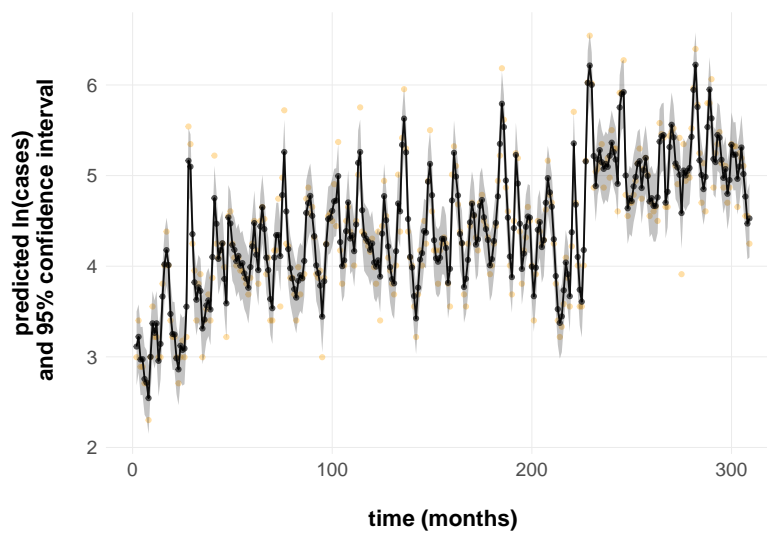


Figure 3: Predictions: $\ln(\text{cases})$ and confidence interval