Some figures for chapter 1

Haakon Bakka

2023 August 11

Contents

Description

This Rmarkdown produces some of the plots in Chapter 1 of Hrafnkelsson et al.

```
stopifnot(file.exists("Book-ch1-figs.Rmd"))
folder.out = "fig-out/"
dir.create(folder.out, showWarnings = F)
```

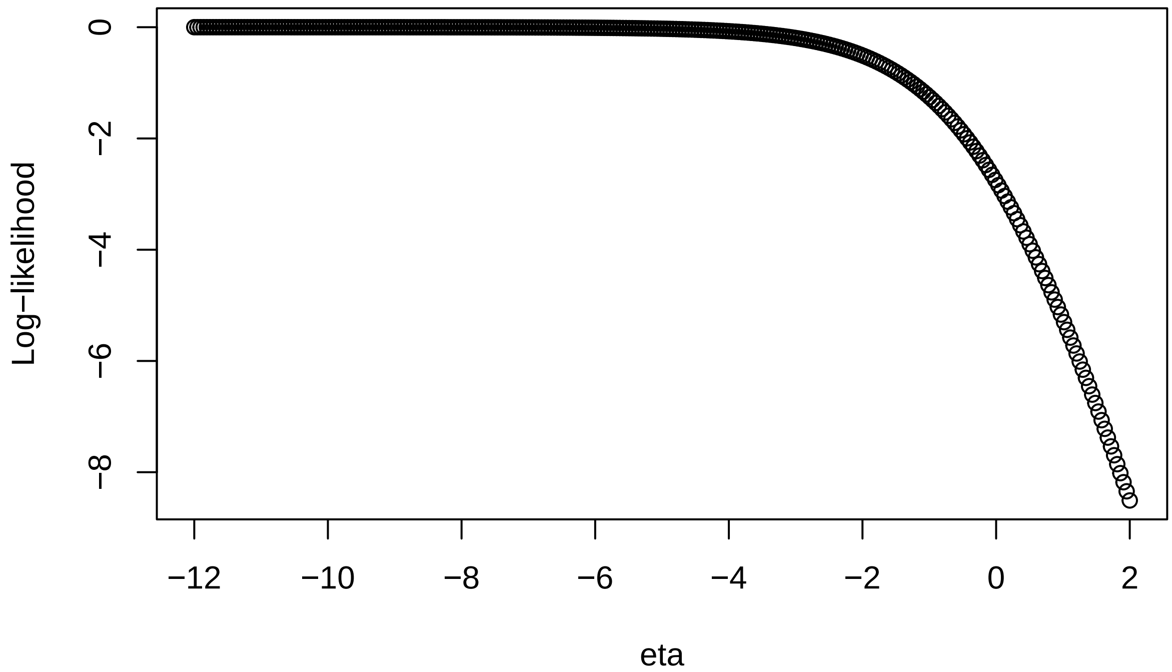
```
library(boot)
library(ggplot2)
library(Matrix)
library(INLA)
```

```
## Loading required package: foreach

## Loading required package: parallel

## Loading required package: sp

## This is INLA_23.04.11-1 built 2023-04-10 23:43:04 UTC.
##  - See www.r-inla.org/contact-us for how to get help.
##  - To enable PARDISO sparse library; see inla.pardiso()
```

Figure 1

The flat likelihoods:

```
library(boot)
etavals = seq(-12, 2, length.out = 300)
gtrue = dbinom(x = 0, size=4, prob=inv.logit(etavals), log = T)
plot(etavals, gtrue, xlab="eta", ylab="Log-likelihood")
```



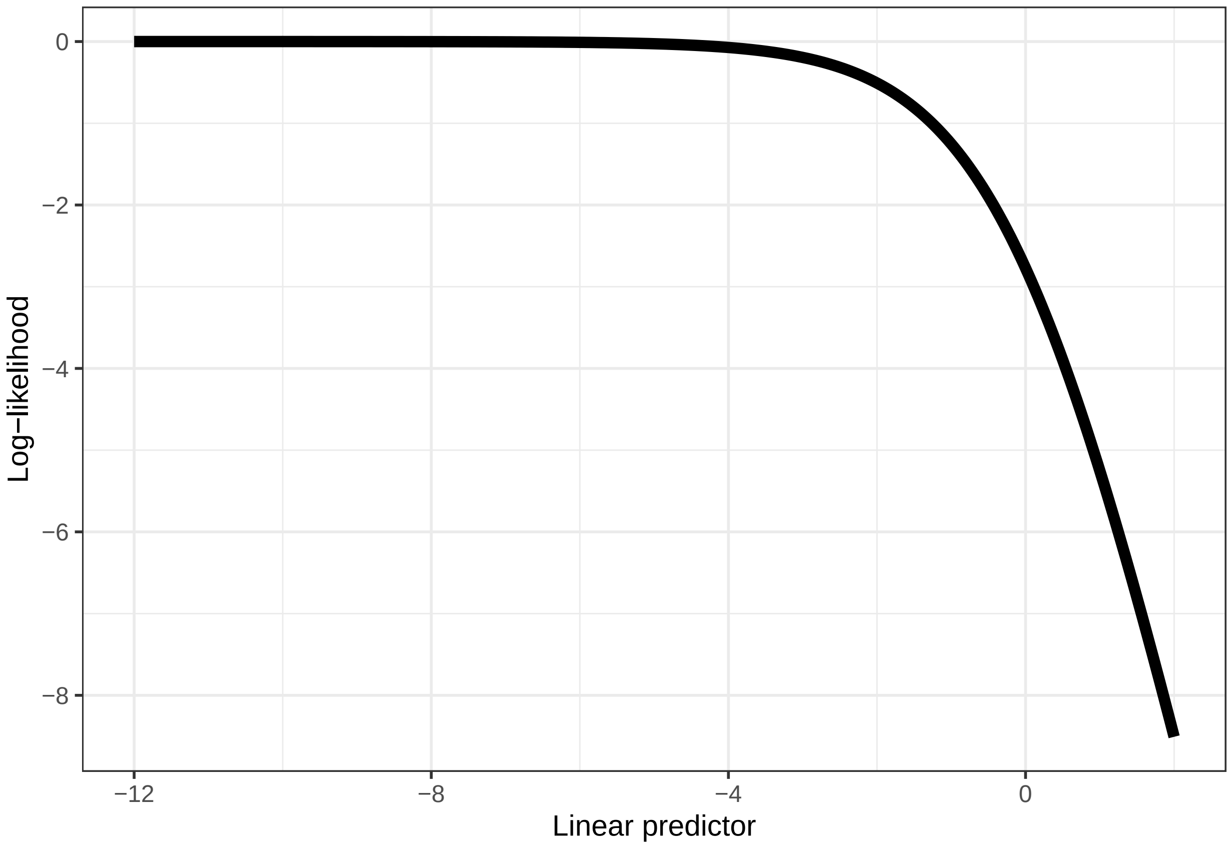

```
png(paste0(folder.out, "likelihood-flat.png"))
etavals = seq(-12, 2, length.out = 300)
gtrue = dbinom(x = 0, size=4, prob=inv.logit(etavals), log = T)
plot(etavals, gtrue, xlab="Linear predictor", ylab="Log-likelihood")
dev.off()
```

```
## pdf
##   2
```

```
## GGPLOT
df = data.frame(etavals, gtrue)
# Basic line plot with points
ggp.lik = ggplot(data=df, aes(x=etavals, y=gtrue))  + theme_bw() +
          geom_line(size=2) + xlab("Linear predictor") + ylab("Log-likelihood")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
ggp.lik
```



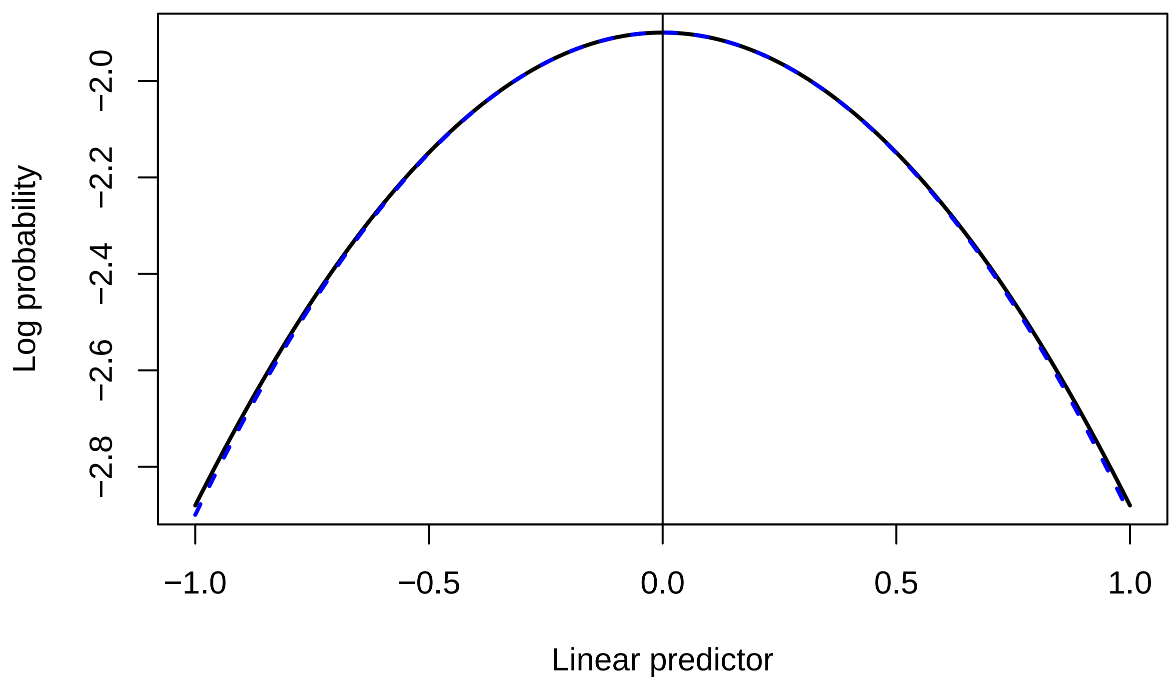

```
ggsave(ggp.lik, filename = paste0(folder.out, "likelihood-flat.png"), height=5, width=7)
```

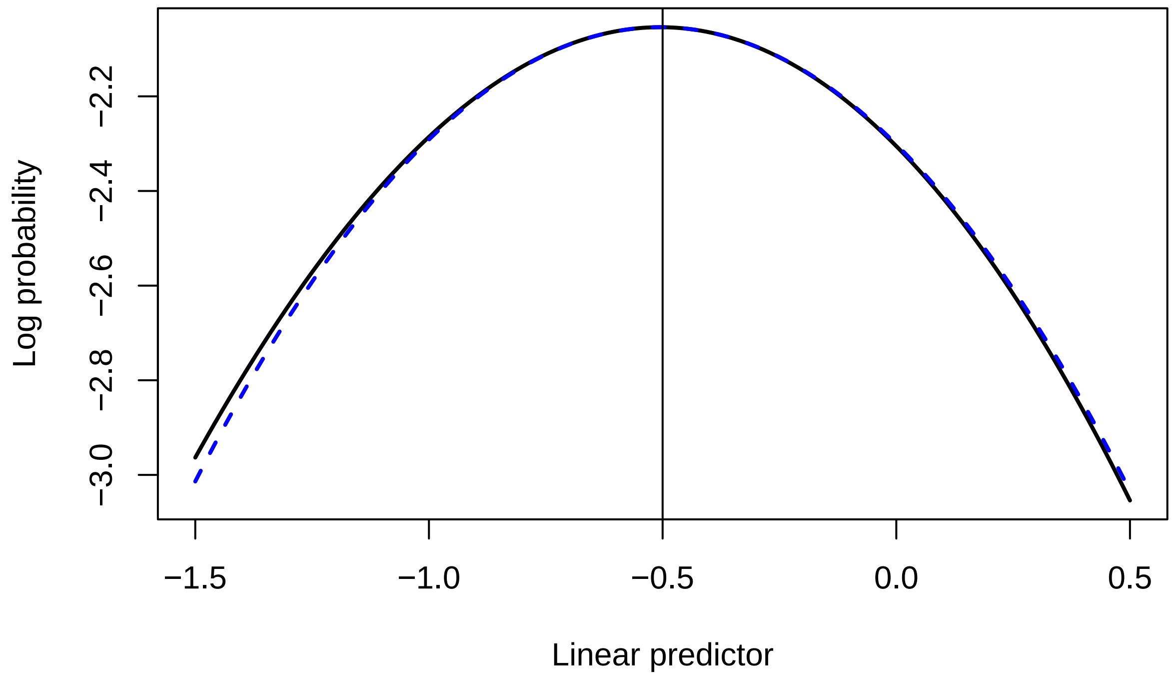
Figure 2

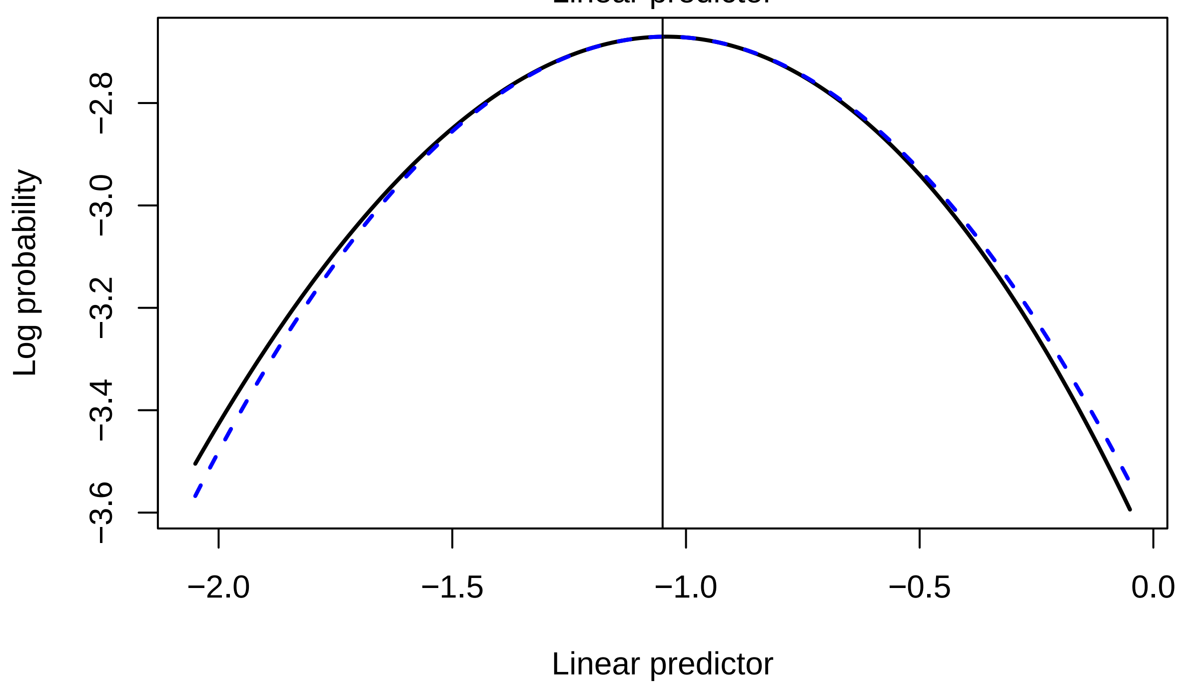
We add a normal(0,1) prior, and do a quadratic approximation at the mode.

```
N.trials = 4
prior.sd = 1
for (i in 1:3) {
  if (i==1) {
    yval = 2
    eta.star = 0 # is the logit(yval/Ntrials) and is the mean of prior
  } else if (i==2) {
    yval = 1
    eta.star = -0.5
  } else {
    yval = 0
    eta.star = -1.05
  }
  p = inv.logit(eta.star)
  lik.d0 = dbinom(x=yval, prob = p, size=N.trials, log=T)
  lik.d1 = yval-N.trials*p
  lik.d2 = - N.trials*p*(1-p)
  a = lik.d0 - eta.star*lik.d1 + 0.5*eta.star^2*lik.d2
  b = lik.d1 - eta.star*lik.d2
  c = -lik.d2
  etavals = seq(-1, 1, length.out = 300) + eta.star
```

```r
  gapprox = a + b*etavals - 1/2 * c *etavals^2
  gtrue = dbinom(x = yval, size=N.trials, prob=inv.logit(etavals), log = T)
  ## Multiply both with the prior
  gapprox = gapprox + dnorm(x=etavals, log = T)
  gtrue = gtrue + dnorm(x=etavals, log = T)
  # = p.and.d12.y.given.eta(eta = eta.star, order = 0)
  plot(etavals, gtrue, type="l", lwd=2, ylab="Log probability", xlab="Linear predictor")
  lines(etavals, gapprox, col="blue", lwd=2, lty="dashed")
  abline(v=eta.star)

  png(paste0(folder.out, "laplace-withg-y",yval, ".png"))
  plot(etavals, gtrue, type="l", lwd=2, ylab="Log probability", xlab="Linear predictor")
  lines(etavals, gapprox, col="blue", lwd=2, lty="dashed")
  abline(v=eta.star)
  dev.off()

}
```







Pick one of the plots and use ggplot

```
## GGplot
df = data.frame(etavals, gtrue, gapprox)
# Basic line plot with points
ggp.lik = ggplot(data=df, aes(x=etavals, y=gtrue))  + theme_bw() +
  geom_line(size=2) +
  #geom_line(aes(y=gapprox), size=2, lty="dotted", col="blue") +
  geom_line(aes(y=gapprox), size=1, col="blue") +
  xlab("Linear predictor") + ylab("Log probability")
ggsave(ggp.lik, filename=paste0(folder.out, "laplace-withg-y",yval, ".png"), height=5, width=7)
ggp.lik
```

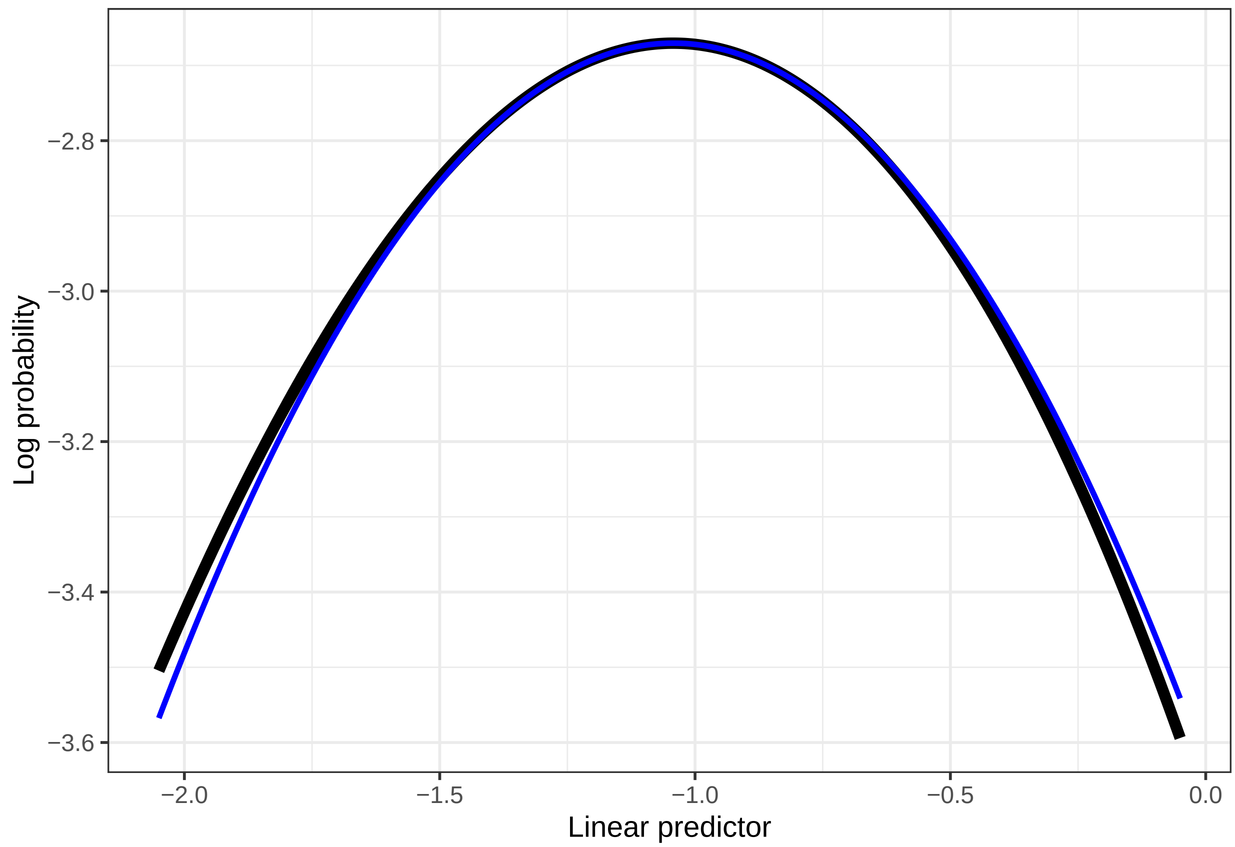


Figure 11

```r
## Note to Birgir: This data is created by the other .rmd file
load("precip.Rdata")
```

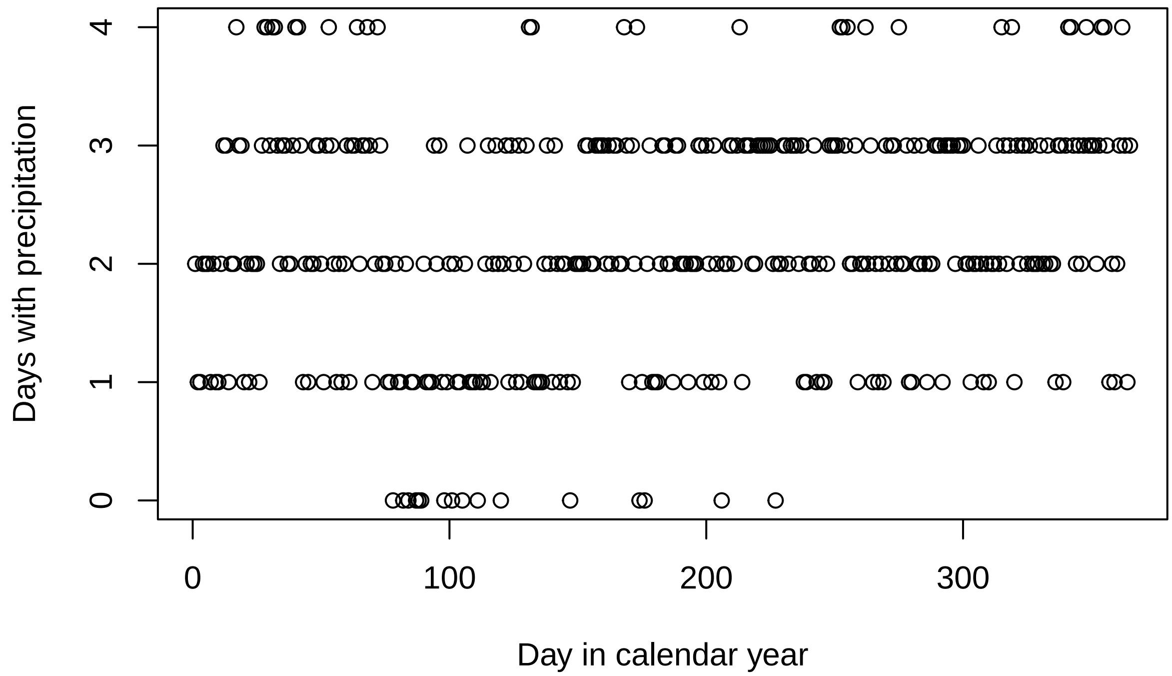
```r
head(precip)
```

```
##   day precip
## 1   1      2
## 2   2      1
## 3   3      1
## 4   4      2
## 5   5      2
## 6   6      2
```

The first column is the day in the year, and the second is a count of how many of the four years where there was precipitation on that day. We deleted the 29th of February year 2020.

We can plot the data as follows.

```r
plot(precip$day, precip$precip, ylab="Days with precipitation", xlab="Day in calendar year")
```




```
png(paste0(folder.out, "precip.png"))
plot(precip$day, precip$precip, ylab="Days with precipitation", xlab="Day in calendar year")
dev.off()
```

```
## pdf
##   2
```

```
## GGPLOT
ggp.precip = ggplot(data=precip, aes(x=day, y=precip))  + theme_bw() +
          geom_point(size=1) + xlab("Day in calendar year") + ylab("Days with precipitation")
ggsave(ggp.precip, filename = paste0(folder.out, "precip.png"), height=5, width=7)
ggp.precip
```

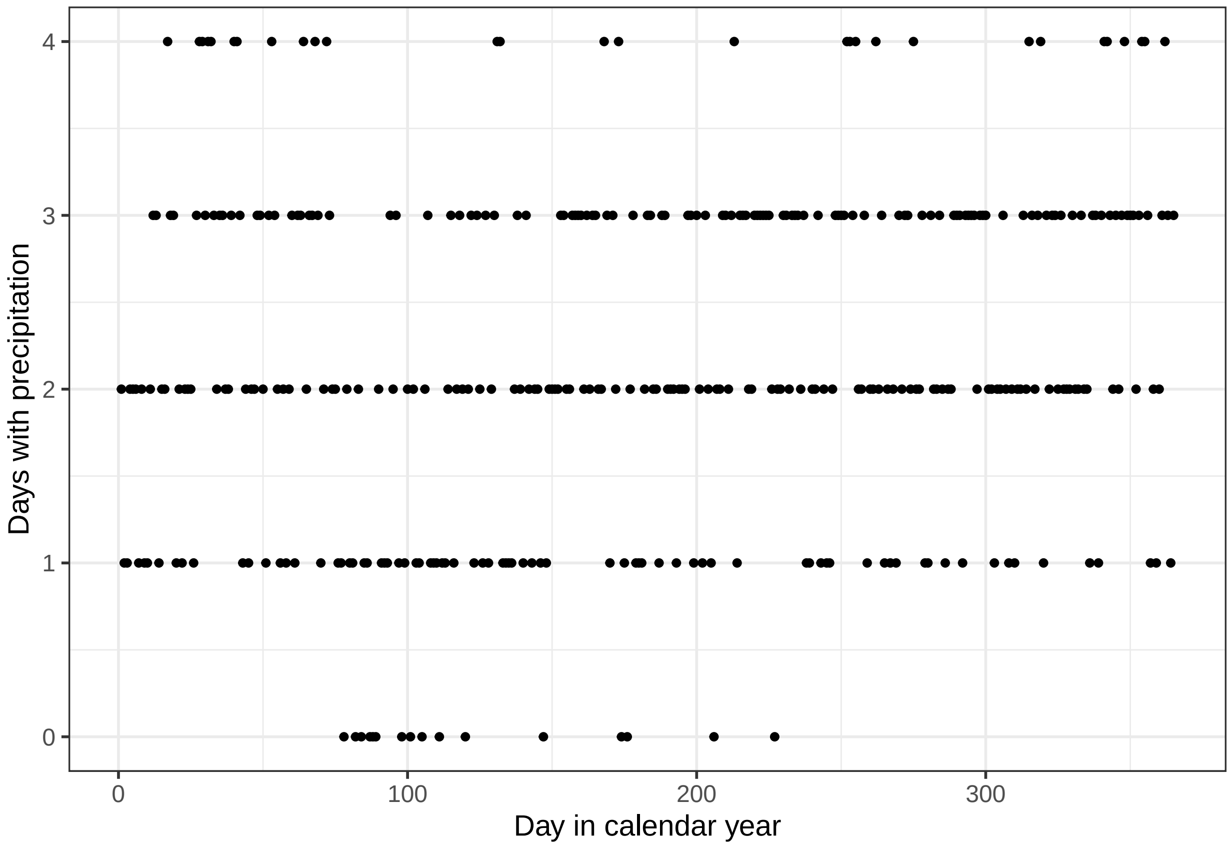
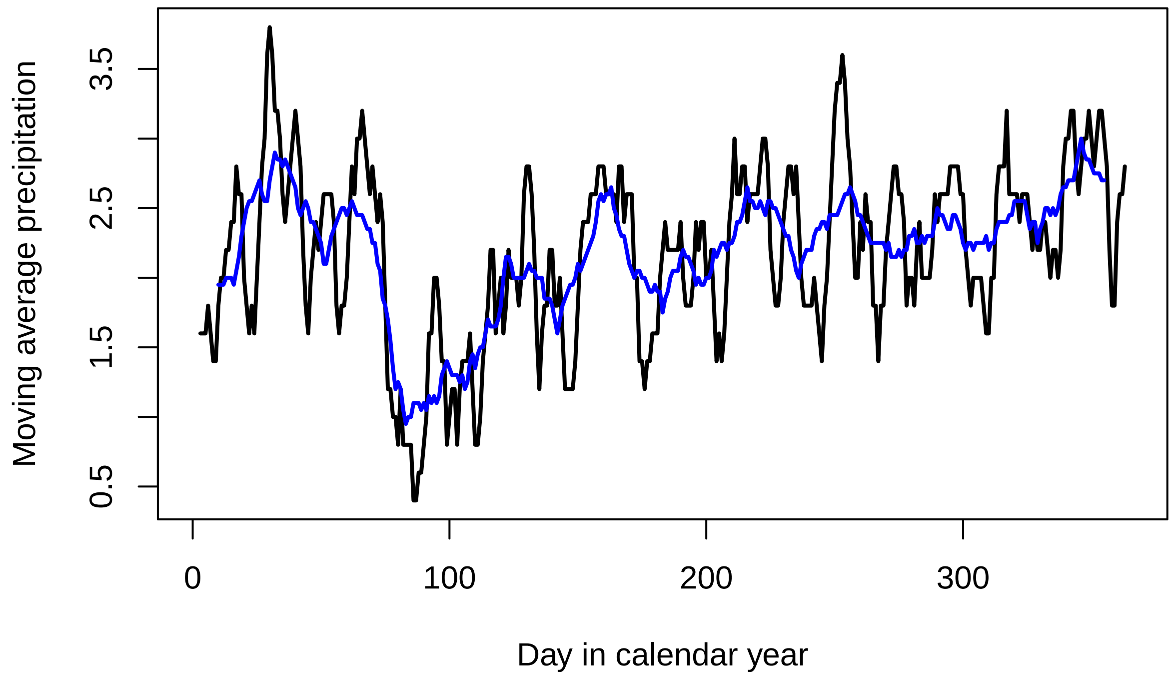


Figure 12

```
ma <- function(x, n = 5){filter(x, rep(1 / n, n), sides = 2)}
m1 = ma(precip$precip, n=5)
m2 = ma(precip$precip, n=20)
plot(m1, ylab="Moving average precipitation", xlab="Day in calendar year", lwd=2)
lines(m2, col="blue", lwd=2)
```




```
png(paste0(folder.out, "precip-ma.png"))
plot(m1, ylab="Moving average precipitation", xlab="Day in calendar year", lwd=2)
lines(m2, col="blue", lwd=2)
dev.off()
```

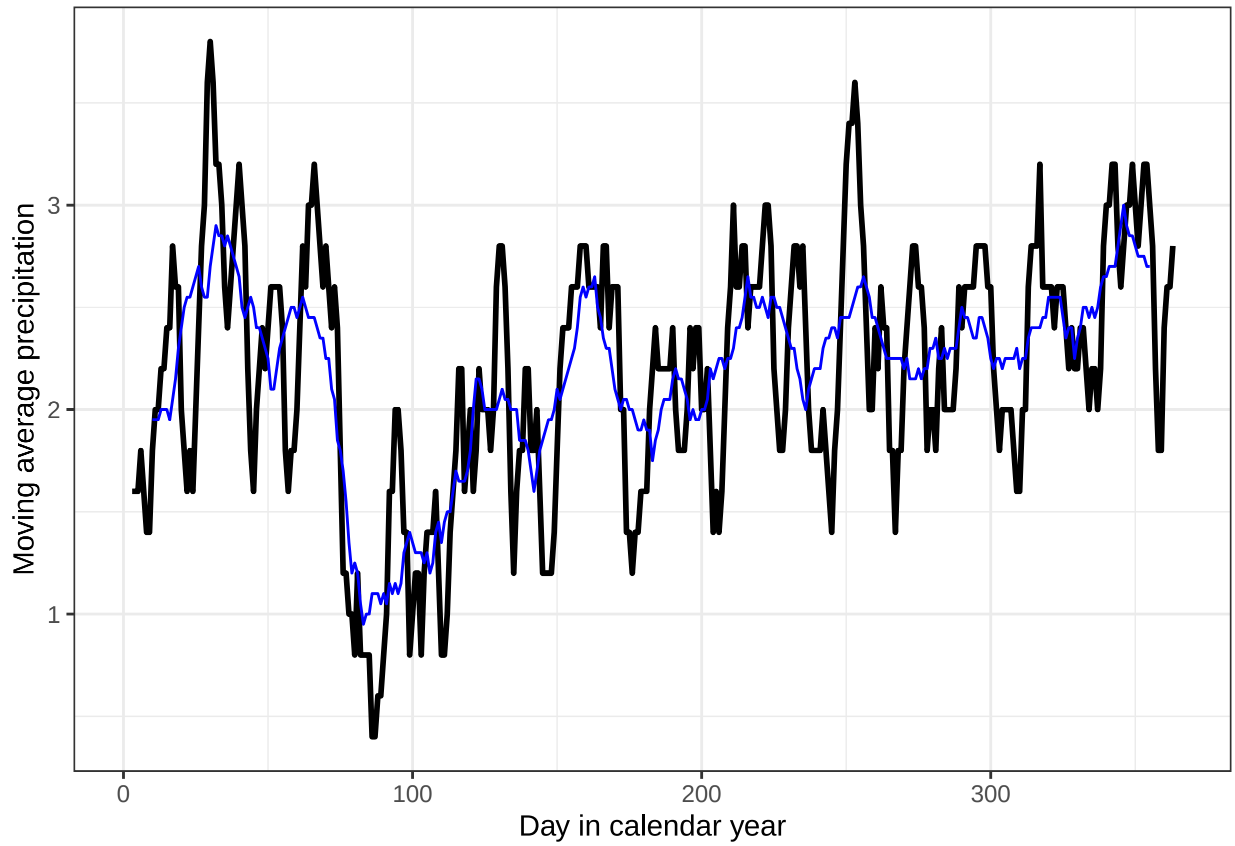
```
## pdf
##   2
```

```
## GGPLOT
df2 = data.frame(day=1:365, m1, m2)
ggp.precip = ggplot(data=df2, aes(x=day, y=m1))  + theme_bw() +
          geom_line(size=1) +
          geom_line(aes(y=m2), col="blue") +
  xlab("Day in calendar year") + ylab("Moving average precipitation")
ggsave(ggp.precip, filename = paste0(folder.out, "precip-ma.png"), height=5, width=7)
```

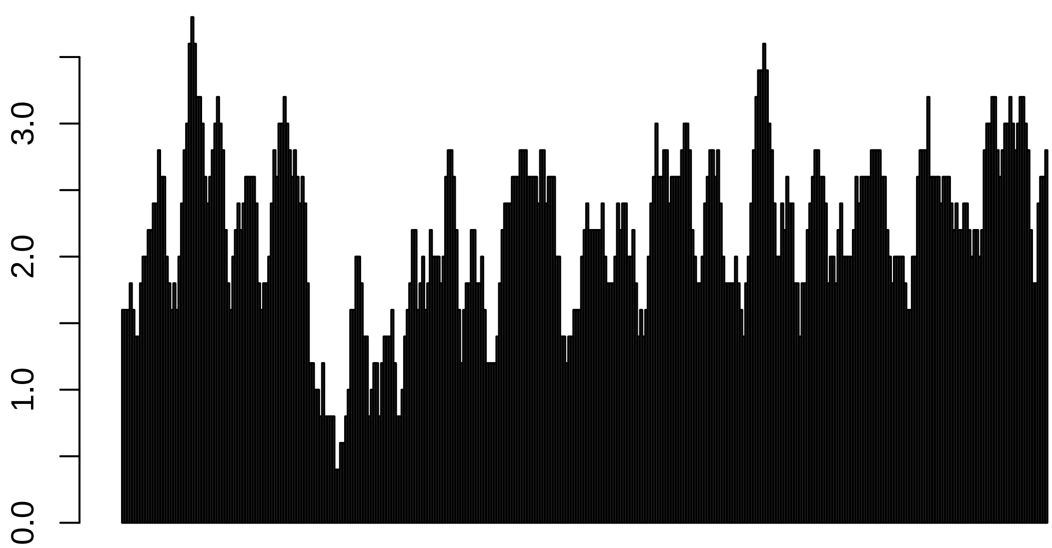
```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.

## Warning: Removed 4 rows containing missing values (`geom_line()`).

## Warning: Removed 19 rows containing missing values (`geom_line()`).
```

```
ggp.precip
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.

## Warning: Removed 4 rows containing missing values (`geom_line()`).
## Removed 19 rows containing missing values (`geom_line()`).
```




```
barplot(as.numeric(m1))
```




```
barplot(as.numeric(m2))
```

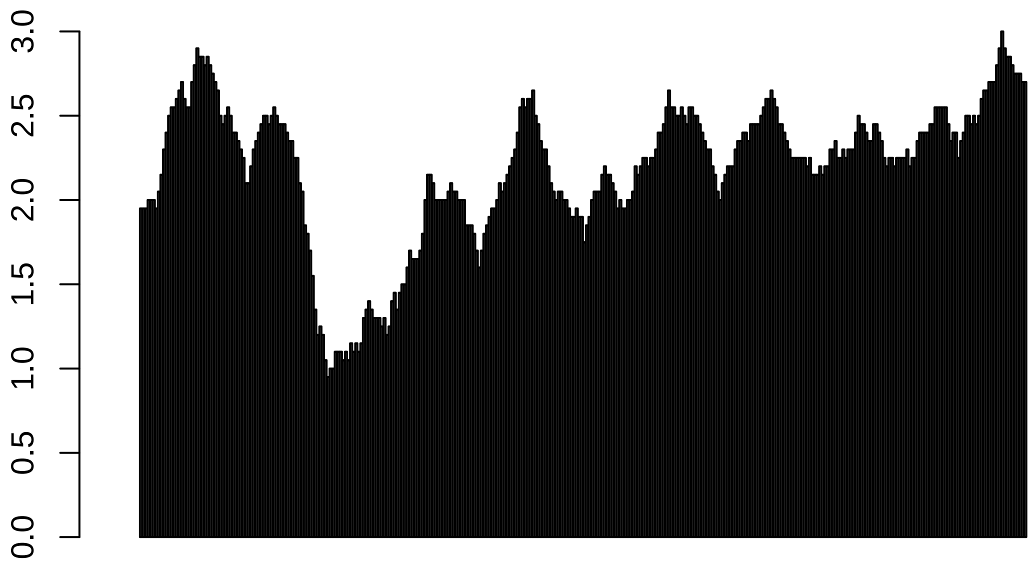


Figure 13

A circular auto-regressive model component.

```
N = nrow(df2)
#N = 10
## For example with
rho = 0.95
Ui = sparseMatrix(1:N, 1:N, x = rep(1, N), dims = c(N,N))
Uim1 = sparseMatrix(2:N, 1:(N-1), x = rep(1, N-1), dims = c(N,N))
T1 = Ui - rho*Uim1
T2 = T1 %*% T1
Q2 = t(T2) %*% T2
```

```
## Make circular
Q2[1, N] = Q2[1, 2]
Q2[1, N-1] = Q2[1, 3]
Q2[2, N] = Q2[2, 4]
Q2[N, 1] = Q2[1, N]
Q2[N-1, 1] = Q2[1, N-1]
Q2[N, 2] = Q2[2, N]

#Q2[1:2, 1:2] = Q2[N - 0:1, N - 0:1]
Q2[N - 0:1, N - 0:1] = Q2[1:2, 1:2]

if(N<11) print(round(Q2, 3))
```

```
## Fix variance problem
range(diag(solve(Q2)))
```

```
## [1] 2052.631 2052.631
```

```
phi1 = 2*rho
phi2 = -rho^2
mvar = (1-phi2)/((1+phi2)*(1-phi1-phi2)*(1+phi1-phi2))
Q3 = Q2 * mvar
range(diag(solve(Q3)))
```

```
## [1] 1 1
```

```r
const.corr = mean(diag(solve(Q2)))
Q3 = Q2*const.corr
range(diag(solve(Q3)))
```

```
## [1] 1 1
```

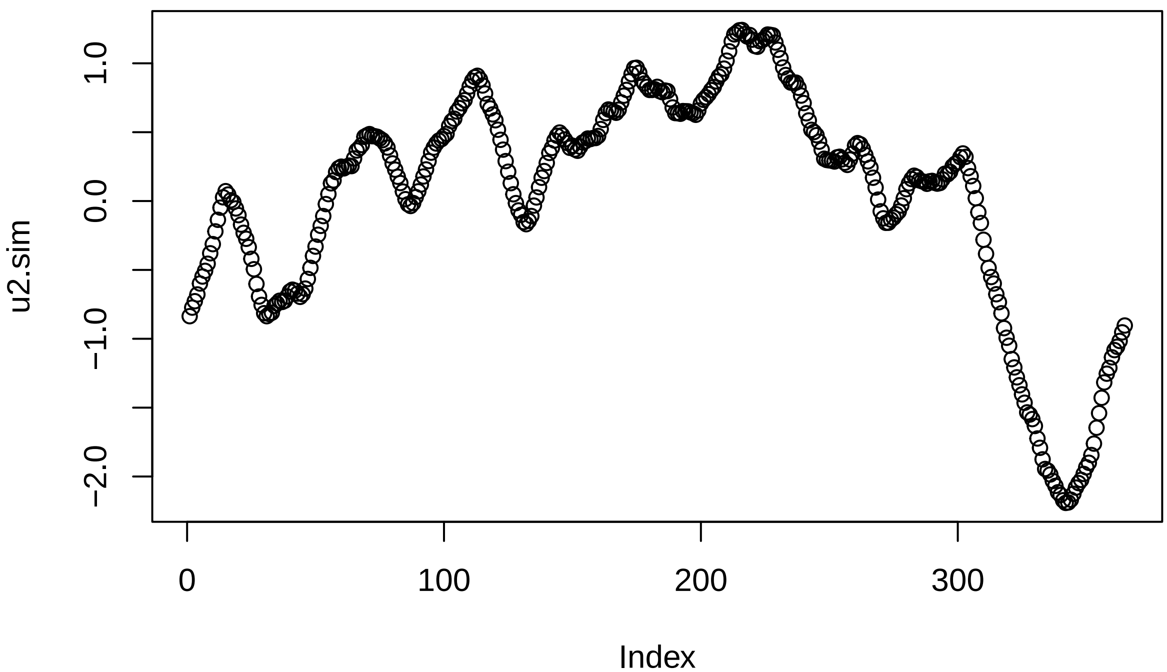
```r
## Correlation
Sig3 = solve(Q3)
Sig3[1, 91]
```

```
## [1] 0.05550826
```

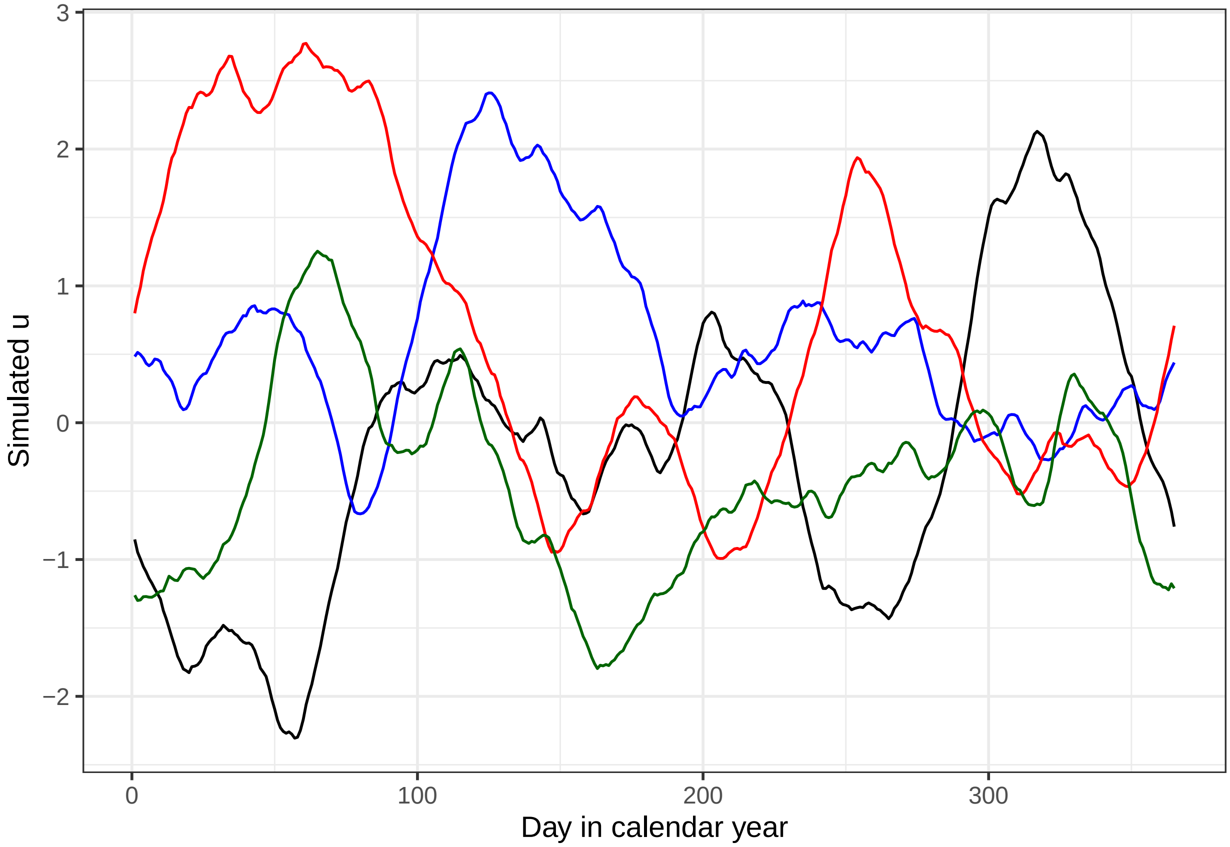
```r
Sig3[90, 90]
```

```
## [1] 1
```

```r
T2 = chol(Q3)
u2.sim = solve(T2, rnorm(N))
plot(u2.sim)
```




```r
set.seed(202106)

dftemp = as.data.frame(as.matrix(solve(T2, cbind(rnorm(N), rnorm(N), rnorm(N), rnorm(N), rnorm(N), rnorr
dftemp$day = df2$day
ggp.sim = ggplot(data=dftemp, aes(x=day, y=V1)) + theme_bw() +
          geom_line(aes(y=V5), col="black") +
          geom_line(aes(y=V2), col="blue") +
          geom_line(aes(y=V3), col="red") +
          geom_line(aes(y=V4), col="darkgreen") +
          xlab("Day in calendar year") + ylab("Simulated u")
ggp.sim
```




```
ggsave(ggp.sim, filename = paste0(folder.out, "sim-u.png"), height=5, width=7)
```

```
Qar2c = Q3
```

Figure 14: Inference

We will use the R-package INLA to perform inference with the Laplace approximation.

```
library(INLA)
hyper1 = list(theta = list(prior="pc.prec", param=c(0.1,0.5)))
## Choose formula
if (T) {
form1 = precip ~ 1 + f(day, model="generic0", Cmatrix = Qar2c, hyper=hyper1)
} else {
  ## Old version
form1 = precip ~ 1 + f(day, model="rw2", constr=TRUE, cyclic=TRUE, scale.model=TRUE, hyper=hyper1)
}
```

We removed the intercept since this is undetermined in the RW2 model (it is in the null space). We also used the cyclic effect $u_1 = u_{365}$

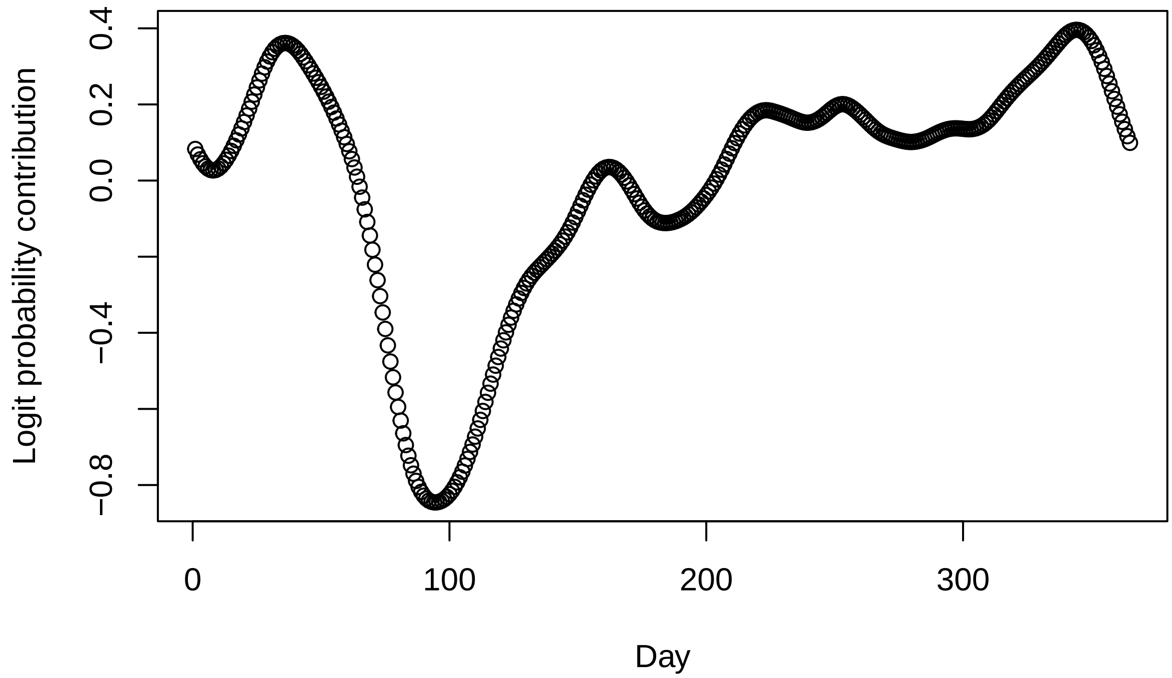
```
fit1 = inla(form1, data=precip, family="binomial", Ntrials = 4,
            control.predictor=list(compute=T),
            control.inla = list(int.strategy="eb", strategy="gaussian"))
```

```
summary(fit1)
```

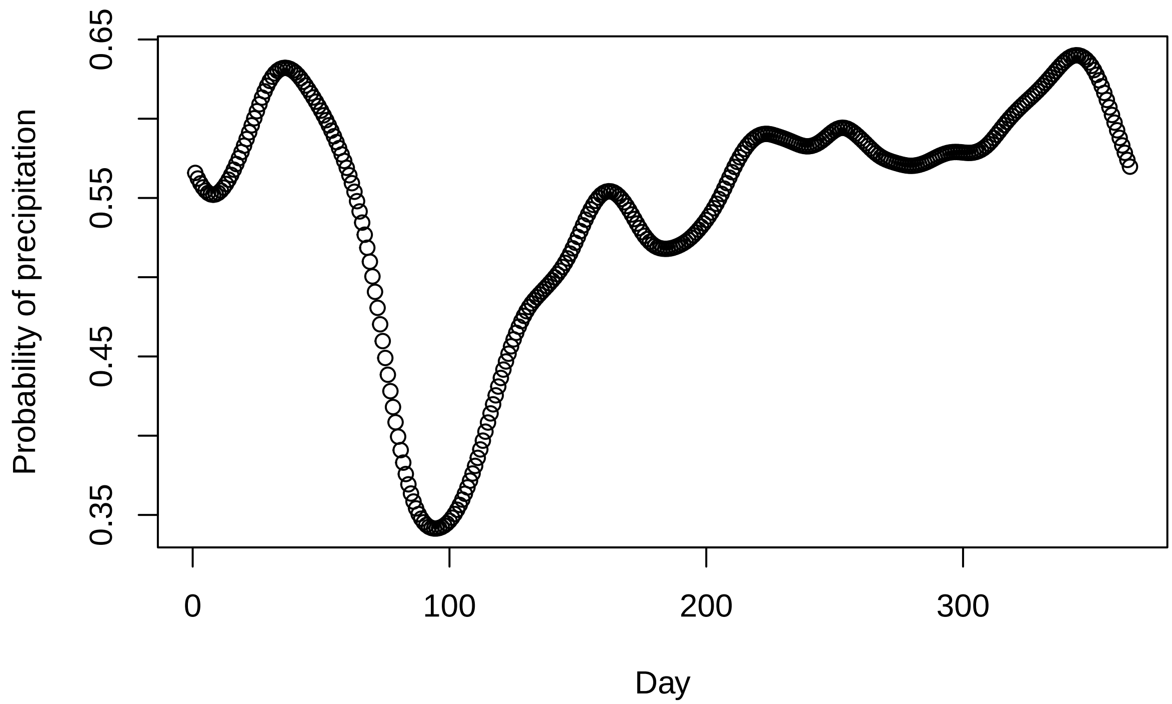
```
##
## Call:
```

```
##    c("inla.core(formula = formula, family = family, contrasts = contrasts,
##    ", " data = data, quantiles = quantiles, E = E, offset = offset, ", "
##    scale = scale, weights = weights, Ntrials = Ntrials, strata = strata,
##    ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose =
##    verbose, ", " lincomb = lincomb, selection = selection, control.compute
##    = control.compute, ", " control.predictor = control.predictor,
##    control.family = control.family, ", " control.inla = control.inla,
##    control.fixed = control.fixed, ", " control.mode = control.mode,
##    control.expert = control.expert, ", " control.hazard = control.hazard,
##    control.lincomb = control.lincomb, ", " control.update =
##    control.update, control.lp.scale = control.lp.scale, ", "
##    control.pardiso = control.pardiso, only.hyperparam = only.hyperparam,
##    ", " inla.call = inla.call, inla.arg = inla.arg, num.threads =
##    num.threads, ", " blas.num.threads = blas.num.threads, keep = keep,
##    working.directory = working.directory, ", " silent = silent, inla.mode
##    = inla.mode, safe = FALSE, debug = debug, ", " .parent.frame =
##    .parent.frame)")
## Time used:
##     Pre = 3.03, Running = 0.502, Post = 0.0288, Total = 3.56
## Fixed effects:
##              mean  sd 0.025quant 0.5quant 0.975quant  mode kld
## (Intercept) 0.184 0.2     -0.207    0.184      0.575 0.184   0
##
## Random effects:
##   Name     Model
##     day Generic0 model
##
## Model hyperparameters:
##                   mean   sd 0.025quant 0.5quant 0.975quant mode
## Precision for day 7.06 4.73       1.80     5.85      19.46 4.06
##
## Marginal log-Likelihood:  -1894.93
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

```
plot(fit1$summary.random$day$ID, fit1$summary.random$day$mean,
     xlab="Day", ylab="Logit probability contribution")
```




```
plot(1:365, fit1$summary.fitted.values$mean,
     xlab="Day", ylab="Probability of precipitation")
```



###

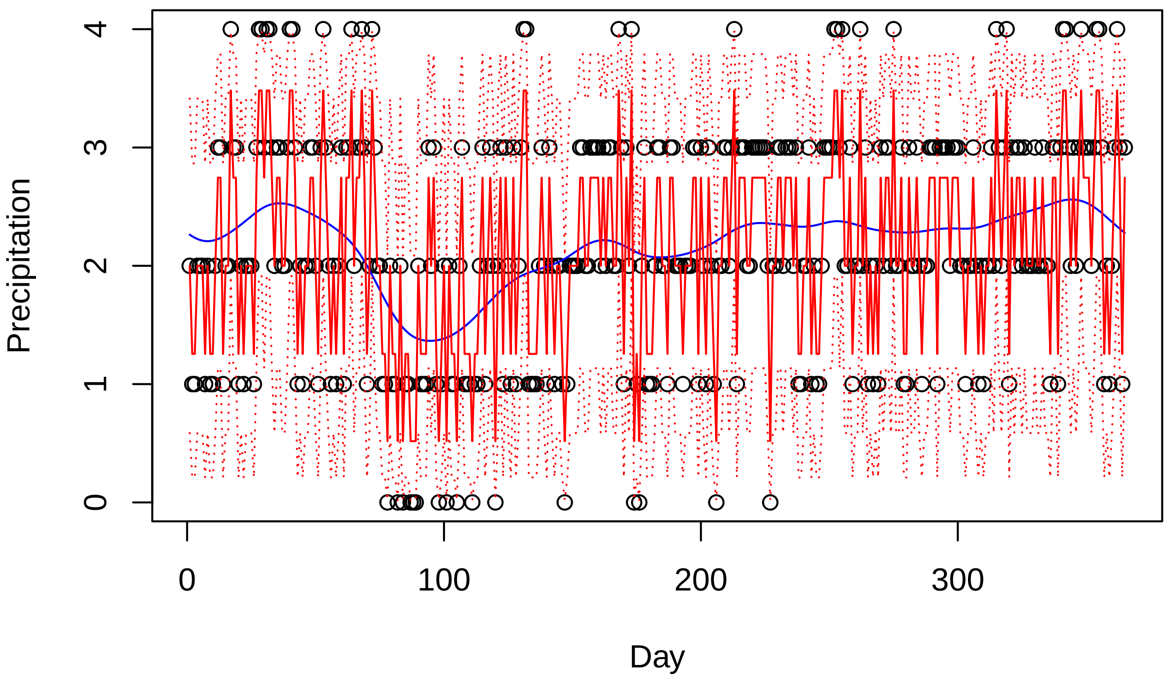
Intermezzo: Individual beta prior

```
## Assume beta(1,1) prior, we get Beta:
post.b1 = 1+precip$precip
post.b2 = 1+4-precip$precip
post.m = post.b1/(post.b1+post.b2)
q1 = qbeta(0.5, post.b1, post.b2)
q2 = qbeta(0.025, post.b1, post.b2)
```

```
q3 = qbeta(0.975, post.b1, post.b2)
q = rbind(q1, q2, q3)
```

Plot together with data: Measured and expected number of events.

```
plot(precip$day, precip$precip, xlab="Day", ylab="Precipitation")
lines(1:365, 4*fit1$summary.fitted.values$mean, col="blue")
lines(1:365, q1*4, col="red")
lines(1:365, q2*4, col="red", lty="dotted")
lines(1:365, q3*4, col="red", lty="dotted")
```



This plot is the goal of this analysis. It shows increased precipitation around day 30, decreased precipitation around day 90, and increased precipitation later in the year.

More advanced approximations

There are two main steps to improving the above approximation. The first is to integrate over the θ posterior. This can be done either with sampling (in MCMC algorithms) or with deterministic integration (in INLA). The second is to improve the estimates of the posterior marginals of u, namely $\pi(u_i|y)$, by another Laplace approximation, see INLA review paper.

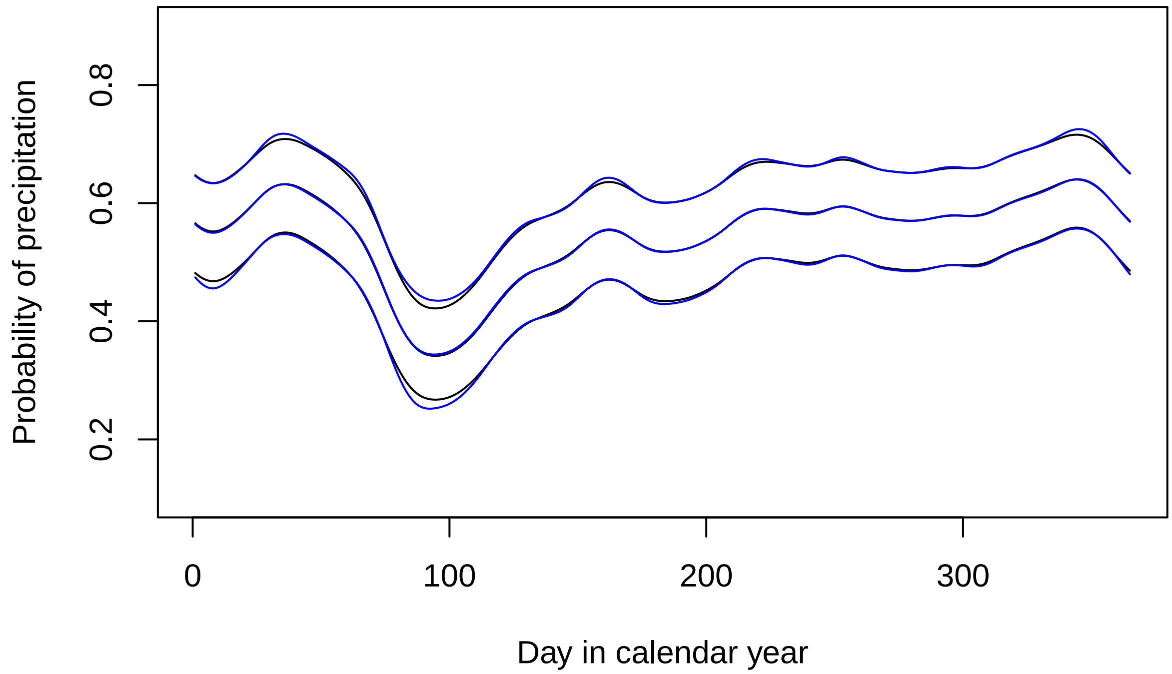
```
fit2 = inla(form1, data=precip, family="binomial", Ntrials = 4,
            control.predictor=list(compute=T))
```

This code is slightly simpler than for `fit1`, because this is the default option in INLA.

Next we compare the old and new results, and the quantiles estimated from the marginals.

```
mult = 1
#ylab1 = "Expected days with precipitation"
ylab1 = "Probability of precipitation"
plot(1:365, mult*fit1$summary.fitted.values$mean, type="l", ylim=c(0.1, 0.9)*mult,
     ylab=ylab1, xlab="Day in calendar year")
lines(1:365, mult*fit1$summary.fitted.values$"0.025quant", type="l")
lines(1:365, mult*fit1$summary.fitted.values$"0.975quant", type="l")
```

```
lines(1:365, mult*fit2$summary.fitted.values$mean, col="blue", cex=0.5)
lines(1:365, mult*fit2$summary.fitted.values$"0.025quant", col="blue", cex=0.5)
lines(1:365, mult*fit2$summary.fitted.values$"0.975quant", col="blue", cex=0.5)
```

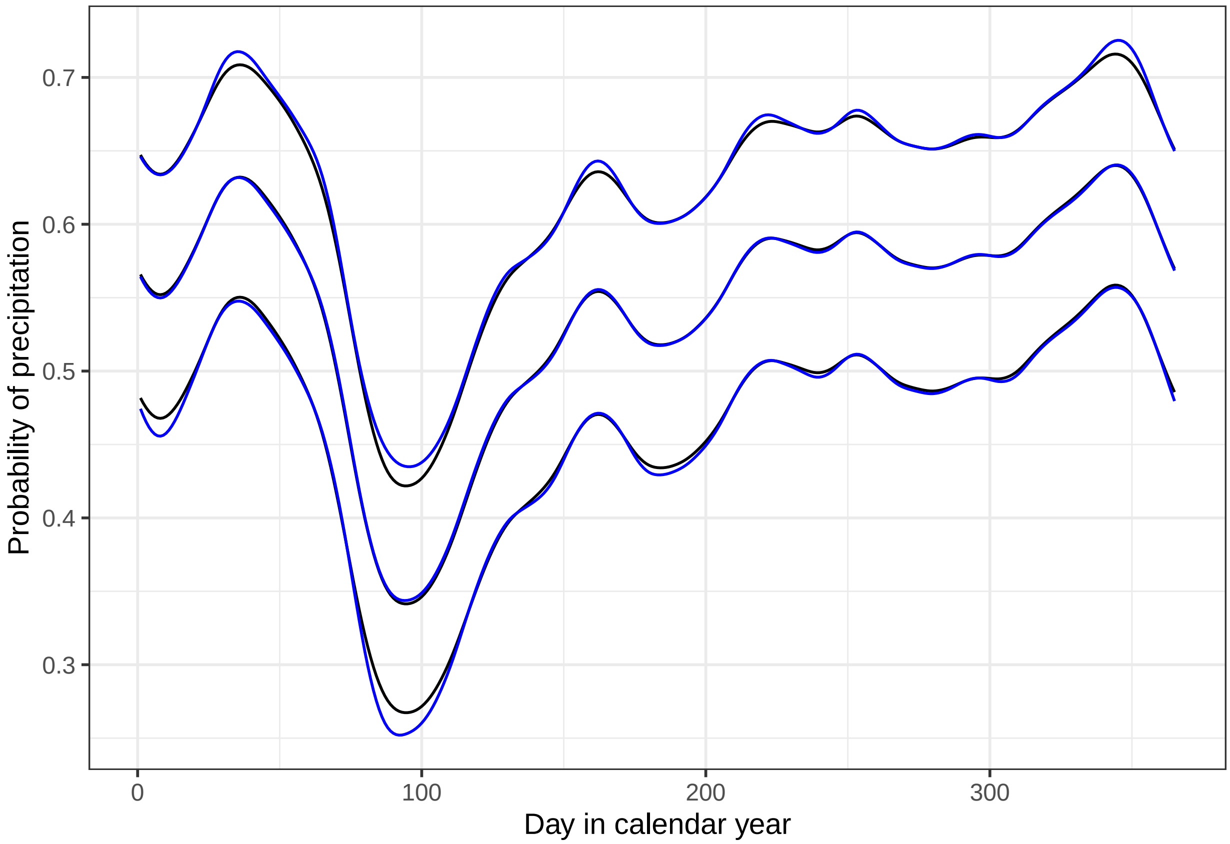



```
#lines(1:365, q1*mult, col="red", type="p")
#lines(1:365, q2*mult, col="red", lty="dotted")
#lines(1:365, q3*mult, col="red", lty="dotted")
#points(precip$day, precip$precip, col="darkgreen")

png(paste0(folder.out, "precip-est2.png"), width = 240*2, height=240*2)
plot(1:365, mult*fit1$summary.fitted.values$mean, type="l", ylim=c(0.2, 0.68)*mult,
     ylab=ylab1, xlab="Day in calendar year")
lines(1:365, mult*fit1$summary.fitted.values$"0.025quant", type="l")
lines(1:365, mult*fit1$summary.fitted.values$"0.975quant", type="l")
lines(1:365, mult*fit2$summary.fitted.values$mean, col="blue", cex=0.5)
lines(1:365, mult*fit2$summary.fitted.values$"0.025quant", col="blue", cex=0.5)
lines(1:365, mult*fit2$summary.fitted.values$"0.975quant", col="blue", cex=0.5)
dev.off()
```

```
## pdf
##   2
```

```
## GGPLOT
df3 = data.frame(day=1:365, e11 = mult*fit1$summary.fitted.values$mean,
                 e12 = fit1$summary.fitted.values$"0.025quant",
                 e13 = fit1$summary.fitted.values$"0.975quant",
                 e21 = fit2$summary.fitted.values$mean,
                 e22 = fit2$summary.fitted.values$"0.025quant",
                 e23 = fit2$summary.fitted.values$"0.975quant")
ggp.precip = ggplot(data=df3, aes(x=day, y=e11)) + theme_bw() +
          geom_line(aes(y=e11)) +
          geom_line(aes(y=e12)) +
          geom_line(aes(y=e13)) +
```

```
        geom_line(aes(y=e21), col="blue") +
        geom_line(aes(y=e22), col="blue") +
        geom_line(aes(y=e23), col="blue") +
        xlab("Day in calendar year") + ylab("Probability of precipitation")

ggsave(ggp.precip, filename = paste0(folder.out, "precip-est.png"), height=5, width=7)
ggp.precip
```



We note that the mean estimate is almost exactly the same, but that the upper and lower quantiles are slightly different. This is because the posterior marginals are slightly skewed, which is not picked up on by the simple quadratic ("Gaussian") approximation.

```
## Day 100, the uncertainties and teh datapoint
## DOnt plot just show all intervals
day = 101
print(precip$precip[day]/4)
```

```
## [1] 0
```

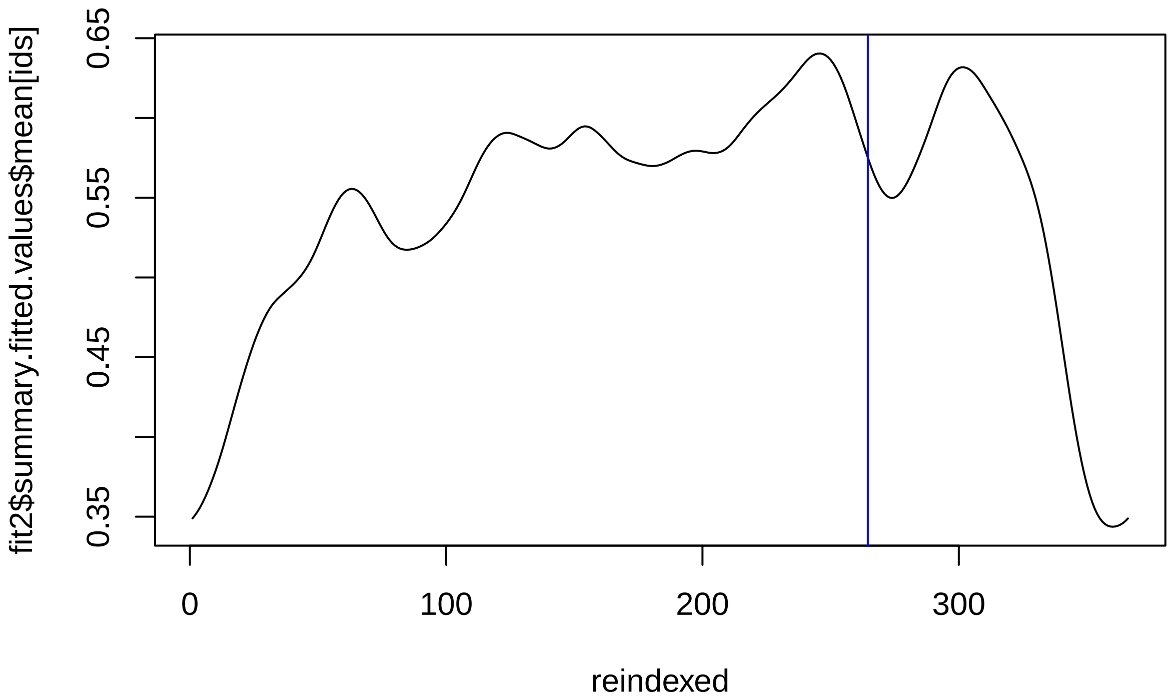
```
f1s = c(fit1$summary.fitted.values$mean[day],
  fit1$summary.fitted.values$"0.025quant"[day],
  fit1$summary.fitted.values$"0.975quant"[day])
f2s = c(fit2$summary.fitted.values$mean[day],
  fit2$summary.fitted.values$"0.025quant"[day],
  fit2$summary.fitted.values$"0.975quant"[day])
a = data.frame(observ = c(precip$precip[day]/4, NA, NA),
               single=q[, day], laplace = f1s, inla = f2s)
rownames(a) = c("Median", "Lower", "Upper")
```

```
round(a[c(1,3,2), ], 3)
```

```
##        observ single laplace  inla
## Median      0  0.129   0.348 0.351
## Upper      NA  0.522   0.429 0.439
## Lower      NA  0.005   0.273 0.263
```

Check that the model is cyclic

```
ids = c(100:N, 1:100)
plot(fit2$summary.fitted.values$mean[ids], type="l", xlab="reindexed")
abline(v=N-100.5, col="blue")
```



End