BAYESIAN INFERENCE USING INLA

SAAVEDRA, P.

```
library(readxl)
library(INLA)
library(quantreg)
library(lme4)
library(tidyverse)
library(flextable)
library(plot3D)
library(epibasix)
library(janitor)
library(lubridate)
library(knitr)
library(kableExtra)
inla.setOption(short.summary=TRUE)
```

Course materials. The data files and R code used in this course can be downloaded from this link.

1. A simple linear regression model

For a dataset $(x_i, y_i) : i = 1, ..., n$, we consider the linear regression model: $y_i \sim N(\mu_i, \sigma)$, being

$$\mu_i = \beta_0 + \beta_1 . x_i$$

We estimate this model in the bayesian framework supposing that $\beta_0 \sim N(0; 10^4)$ and $\beta_1 \sim N(3, 1)$

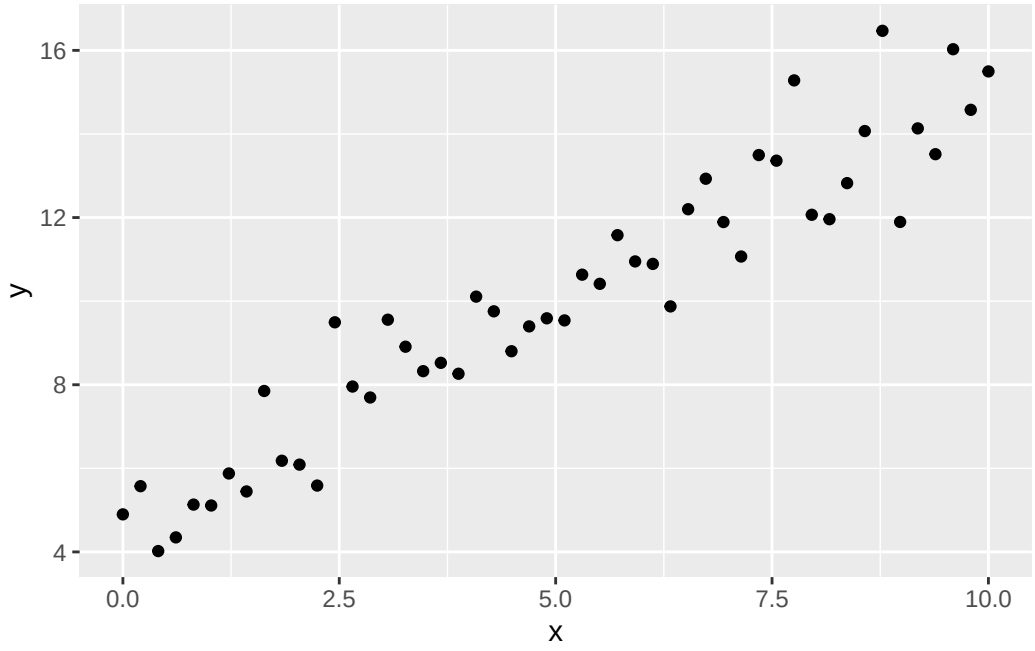
First, we simulate a dataset $\{(x_i, y_i) \ : \ i = 1, ..., n = 50\}$ obeying the linear regression model:

$$y_i \sim N(5 + x_i \ ; \ \sigma = 1)$$

Note $\beta_0 = 5$ and $\beta_1 = 1$.

Data simulation:

```r
### Data simulation
set.seed(19941998)
b <- c(5,1)
sg=1
n=50
x <- seq(0,10,length=50)
mu <- b[1]+b[2]*x
y <- mu+rnorm(n,0,sg)
dt <- data.frame(x,y)
### Figure
ggplot(dt,aes(x,y)) + geom_point()
```



Results of frequentist inference are:

```r
### Frequentist Inference
ml <- lm(y ~x)
summary(ml)
```

```
Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-2.3367 -0.6464 -0.1080  0.6322  2.4538

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.66664    0.28812   16.20   <2e-16 ***
x            1.06518    0.04965   21.45   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.034 on 48 degrees of freedom
Multiple R-squared:  0.9056,    Adjusted R-squared:  0.9036
F-statistic: 460.3 on 1 and 48 DF,  p-value: < 2.2e-16
```

And results of bayesian inference with vague priors:

```
#### Bayesian inference
formula <- y ~ 1 + x
ml <- inla(formula,family="gaussian",data=dt)
ml$summary.fixed[,1:5]
```

```
                mean         sd 0.025quant 0.5quant 0.975quant
(Intercept) 4.666657 0.28746911  4.1006219 4.666656   5.232693
x           1.065182 0.04953899  0.9676378 1.065182   1.162725
```

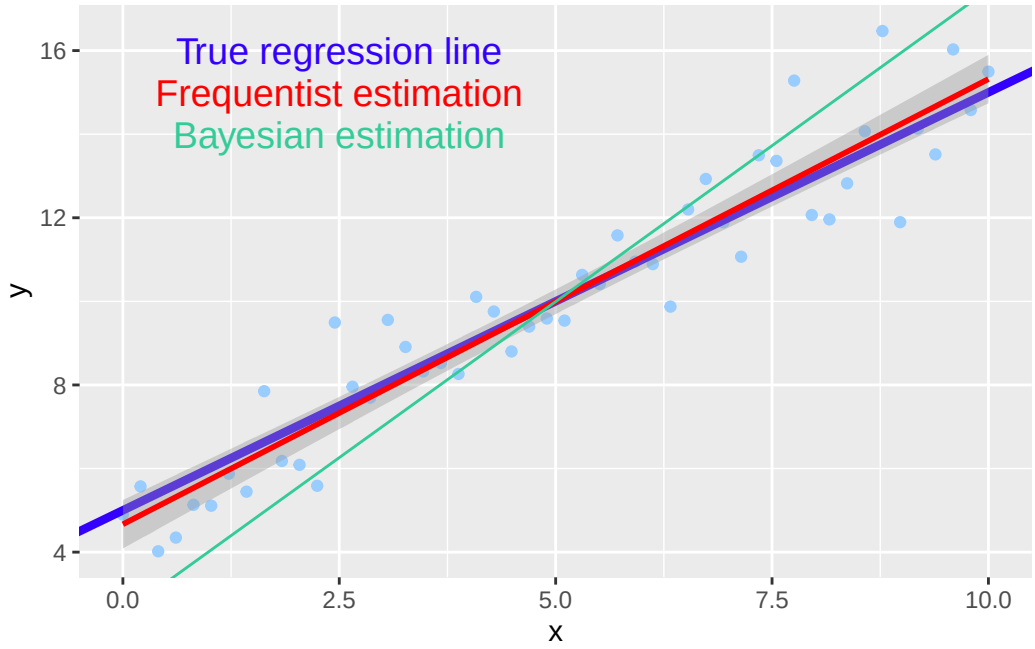
Erratic prior. If we impose on the Bayesian estimation that we are very sure that the value of the slope is 1.5, we arrive at the following result:

```
#### Erratic priors
mb <- inla(formula,family="gaussian",data=dt,
           control.fixed=list(mean=1.5, prec=10000,
                              mean.intercept=1, prec.intercept=0.0001))
mb$summary.fixed[,1:5]
```

```
                mean         sd 0.025quant 0.5quant 0.975quant
(Intercept) 2.528839 0.23580079   2.062869 2.529498   2.991116
x           1.492742 0.01001538   1.473101 1.492742   1.512385
```

Next we show this line as well as the true regression line and the one fitted by frequentist estimation. We can see how our initial prior has had a great effect.

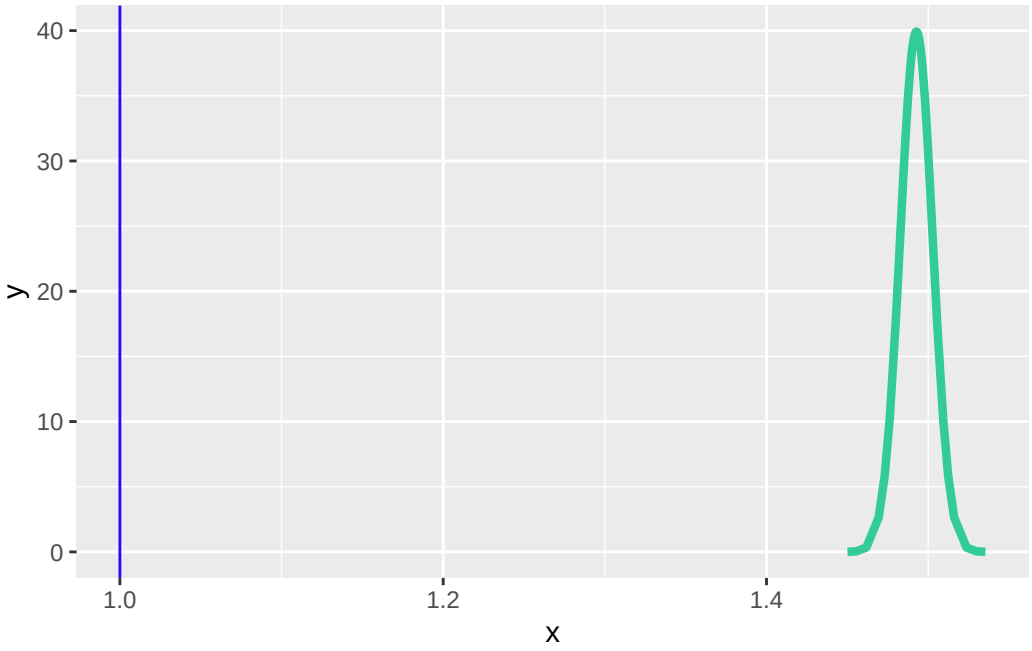
```
### Figure
Bb_0 <- mb$summary.fixed[1,1]
Bb_1 <- mb$summary.fixed[2,1]
ggplot(dt,aes(x,y))+
  geom_point(colour="#99CCFF")+
  geom_abline(intercept = 5, slope = 1,colour="#3300FF",linewidth=1.5)+
  annotate(geom="text", x=2.5, y=16, label="True regression line",colour="#3300FF",size=5)+
  stat_smooth(method = "lm",formula = y ~ x,geom = "smooth",colour="#FF0000")+
  annotate(geom="text", x=2.5, y=15, label="Frequentist estimation",colour="#FF0000",size=5)+
  geom_abline(intercept = Bb_0, slope = Bb_1,colour="#33CC99")+
  annotate(geom="text", x=2.5, y=14, label="Bayesian estimation",colour="#33CC99",size=5)
```



We can also see the posterior density resulting from our insistence that the slope has to be very close to 1.5:

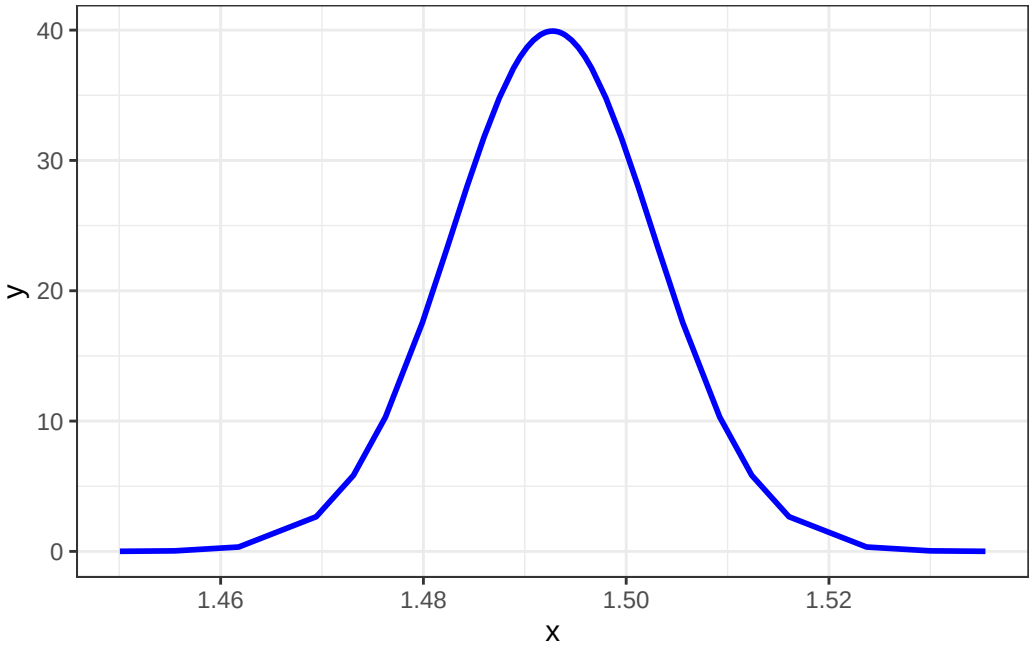
```
#### Densidades a posteriori
dpost.intercept <- data.frame(mb$marginals.fixed$`(Intercept)`)
dpost.x <- data.frame(mb$marginals.fixed$x)

ggplot(dpost.x,aes(x,y))+
```

```
  geom_line(colour="#33CC99",linewidth=1.5)+
  geom_vline(xintercept=1,colour="#3300FF")
```



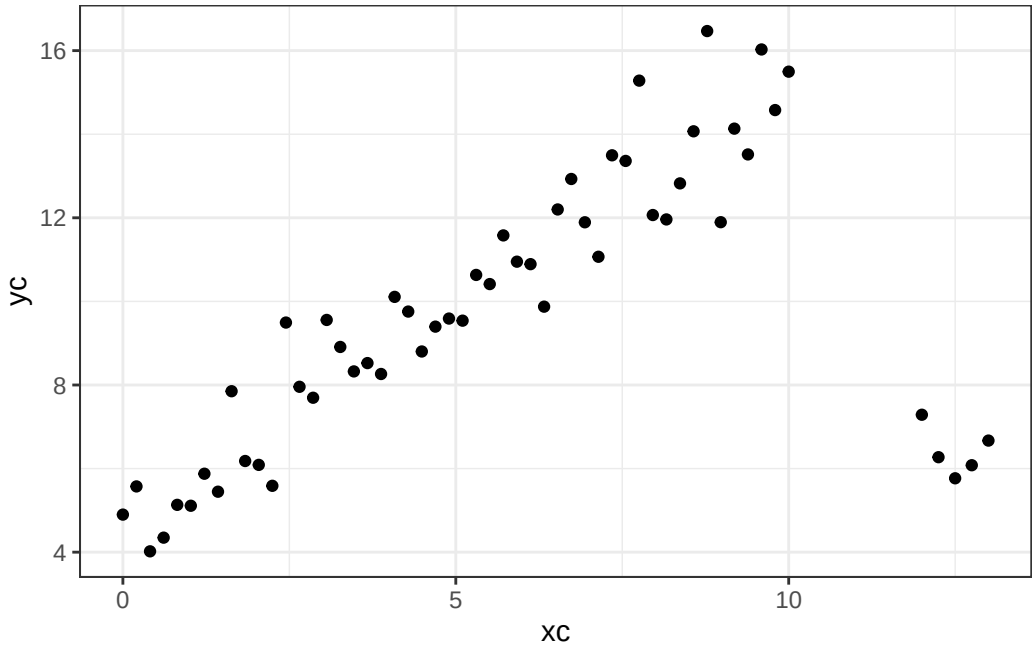
A closer look at the posterior density:

```
ggplot(dpost.x,aes(x,y))+
  geom_line(colour="blue",size=1)+
  theme_bw()
```



Contaminated data. Let's first simulate some contaminated data for a linear regression. We have the same data as before plus a new set of observations that departs markedly from the rest:

```
xz <- seq(12,13,length=5)
yz <- 0.5*xz+1+rnorm(5,0,1)
xc <- c(x,xz)
yc <- c(y,yz)
dc <- data.frame(xc,yc)

ggplot(dc,aes(xc,yc)) +
  geom_point() +
  theme_bw()
```



The frequentist estimation in this case is:

```
ml2 <- lm(yc~xc)
summary(ml2)
```

```
Call:
lm(formula = yc ~ xc)

Residuals:
    Min      1Q  Median      3Q     Max
-7.2558 -1.5951  0.4323  1.7201  5.2777

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.8705     0.7308   9.401 6.97e-13 ***
xc            0.4922     0.1092   4.510 3.63e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.867 on 53 degrees of freedom
Multiple R-squared:  0.2773,    Adjusted R-squared:  0.2637
F-statistic: 20.34 on 1 and 53 DF,  p-value: 3.631e-05
```

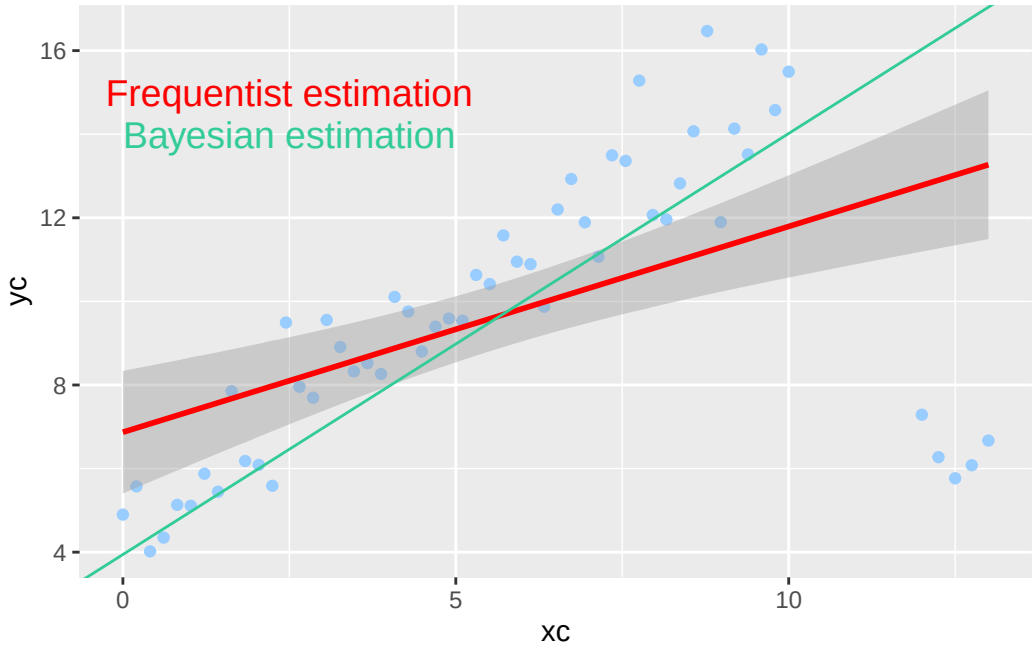
And the bayesian estimation when we consider that the intercept is close to 1.5:

```
formula <- yc ~ 1 + xc
mb <- inla(formula,family="gaussian",data=data.frame(xc,yc),
           control.fixed=list(mean=1.5, prec=64,
           mean.intercept=1, prec.intercept=0.0001))
mb$summary.fixed[,1:5]
```

```
               mean        sd 0.025quant 0.5quant 0.975quant
(Intercept) 3.942956 0.7520936  2.4072531 3.964415   5.358639
xc          1.007470 0.1049289  0.8085617 1.005091   1.219561
```

Graphically:

```
## Figure
Bb_0 <- mb$summary.fixed[1,1]
Bb_1 <- mb$summary.fixed[2,1]
ggplot(dc,aes(xc,yc))+
  geom_point(colour="#99CCFF")+
  stat_smooth(method = "lm",formula = y ~ x,geom = "smooth",colour="#FF0000")+
  annotate(geom="text", x=2.5, y=15, label="Frequentist estimation",colour="#FF0000",size=5)+
  geom_abline(intercept = Bb_0, slope = Bb_1,colour="#33CC99")+
  annotate(geom="text", x=2.5, y=14, label="Bayesian estimation",colour="#33CC99",size=5)
```

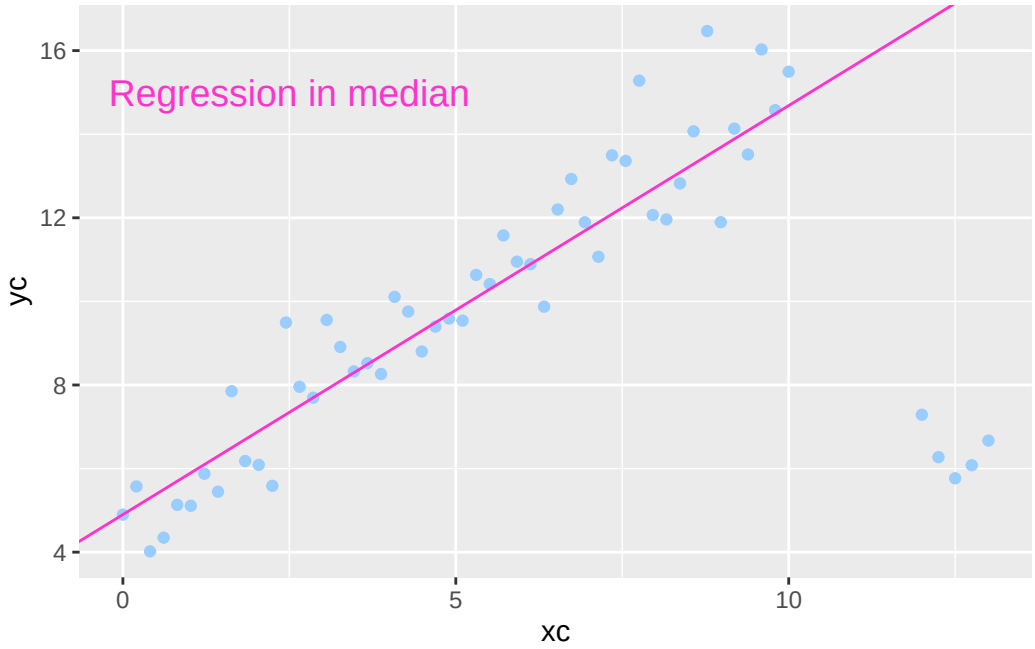


An alternative method when data are contaminated as shown in the figure could be to use regression in median:

```
### Regression in median
rqfit <- rq(yc ~ xc, data =dc)
(b <- rqfit$coeff)
```

```
(Intercept)          xc
   4.898679    0.978623
```

```
ggplot(dc,aes(xc,yc))+
  geom_point(colour="#99CCFF")+
  geom_abline(intercept = b[1], slope = b[2],colour="#FF33CC")+
  annotate(geom="text", x=2.5, y=15, label="Regression in median",colour="#FF33CC",size=5)
```



2. Multivariate linear regression

We now simulate a linear regression model with two independent variables x_1 and x_2. Specifically, the simulated model is $y_i \sim N(\mu_i, \sigma)$ bein $\sigma = 1$ and:

$$\mu_i = 5 + 3x_{1i} - 2x_{2i}$$

```
set.seed(19941998)
b=c(5,3,-2)
sg <- 1
n <- 100
x1 <- rnorm(n,9,1)
x2 <- rnorm(n,5,1)
mu <- b[1]+b[2]*x1+b[3]*x2
y <- mu+rnorm(n,0,sg)
dt <- data.frame(x1,x2,y)
```

Frequentist estimation.

```
# Compute the linear regression
### Maximum Likelihood
ml <- lm(y ~ x1 + x2)
summary(ml)
```

```
Call:
lm(formula = y ~ x1 + x2)

Residuals:
     Min       1Q   Median       3Q      Max
-3.08059 -0.70405  0.02905  0.86390  2.61838

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.5016     1.0417   4.321 3.76e-05 ***
x1            3.0583     0.0982  31.145  < 2e-16 ***
x2           -2.0093     0.1065 -18.864  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.046 on 97 degrees of freedom
Multiple R-squared:  0.9339,    Adjusted R-squared:  0.9326
F-statistic: 685.5 on 2 and 97 DF,  p-value: < 2.2e-16
```

Bayesian estimation with a non informative prior distribution.

```
### Bayes: Non-informative prior distribution
formula <- y ~ 1 + x1 + x2
mb <- inla(formula,family="gaussian",data=dt)
mb$summary.fixed[,1:5]
```

```
                 mean        sd 0.025quant  0.5quant 0.975quant
(Intercept)  4.501754 1.0406963   2.456626  4.501752   6.546890
x1           3.058316 0.0980996   2.865534  3.058316   3.251096
x2          -2.009294 0.1064114  -2.218409 -2.009294  -1.800178
```

```
# Para mostrar los valores de los hiperparámetros:
# mb$all.hyper$fixed
```

Bayesian estimation when researcher has a high confidence that $\beta_0 = 5$.

```
formula <- y ~ 1 + x1 + x2
mb <- inla(formula,family="gaussian",data=dt,
           control.fixed=list(mean=list(x1=0,x2=0),
                              prec=list(x1=0, x2=0),
                              mean.intercept=5, prec.intercept=4))
mb$summary.fixed[,1:5]
```

```
                mean         sd 0.025quant  0.5quant 0.975quant
(Intercept)  4.904802 0.44984447   4.022927  4.904659   5.787485
x1           3.025389 0.06102433   2.905469  3.025432   3.145065
x2          -2.030739 0.09371082  -2.214959 -2.030701  -1.846730
```

```
#mb$all.hyper$fixed
```

Bayesian estimation when researcher has a high confidence that $\beta_1 = 1$ *and* $\beta_2 = -1$.

```
formula <- y ~ 1 + x1 + x2
mb <- inla(formula,family="gaussian",data=dt,
           control.fixed=list(mean=list(x1=1,x2=-1),
                              prec=list(x1=1000, x2=1000),
                              mean.intercept=0, prec.intercept=0))
mb$summary.fixed[,1:5]
```

```
                mean         sd 0.025quant  0.5quant 0.975quant
(Intercept) 17.983424 0.41503532 17.1718441 17.982493 18.8002720
x1           1.034642 0.03172386  0.9724263  1.034642  1.0968563
x2          -1.015382 0.03147097 -1.0770973 -1.015384 -0.9536584
```

```
# mb$all.hyper$fixed
```

3. Logistic regression: association between obesity and low level of education (study of Telde)

Data reading:

```
dt <- read_excel("data/Telde.INLA.xlsx")
dt$Obesity <- ifelse(dt$BMI>=30,1,0)
if (is_html_output()){
  dt %>%
  kbl() %>%
  kable_styling(c("striped", "hover"),full_width = FALSE) %>%
  scroll_box(height = "250px")
} else{
  head(dt) %>% flextable() %>% autofit()
}
```

Age	Sex.Male	BMI	Low.Educational.Level	Obesity	HTA.OMS	DM
44	1	39.30552	1	1	1	0
68	1	24.46460	1	0	0	1
39	1	27.41137	0	0	0	0
49	1	31.40766	0	1	0	0
37	0	29.51594	1	0	0	0
40	1	21.87711	0	0	0	0

Subjects with a low level of education are older than the rest:

```
dt %>%
  mutate(`Low Educational Level`=factor(Low.Educational.Level,
                                        levels=c(0,1),labels=c("No","Yes"))) %>%
  group_by(`Low Educational Level`) %>%
  summarize(`Age (mean±sd)`=sprintf("%.2f±%.2f",mean(Age),sd(Age))) %>%
  flextable() %>%
  autofit()
```

Low Educational Level	Age (mean±sd)
No	44.05±10.01
Yes	55.11±11.82

```
t.test(Age ~ Low.Educational.Level,data=dt)
```

```
    Welch Two Sample t-test

data:  Age by Low.Educational.Level
t = -15.207, df = 668.03, p-value < 2.2e-16
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
95 percent confidence interval:
 -12.485504  -9.629977
sample estimates:
mean in group 0 mean in group 1
       44.05008        55.10782
```

Obesity and low level of education are associated variables:

```
dt %>%
  mutate(`Low Educational Level`=factor(Low.Educational.Level,
                                        levels=c(0,1),labels=c("No","Yes")),
         Obesity=factor(Obesity,levels=0:1,labels=c("No","Yes"))) %>%
  tabyl(`Low Educational Level`,Obesity) %>%
  adorn_percentages("row") %>%
  adorn_pct_formatting() %>%
  adorn_ns("front") %>%
  flextable() %>%
  add_header_row(values=c("","Obesity"),colwidths = c(1,2)) %>%
  vline(j=1) %>%
  vline_left(border=officer::fp_border(color="black")) %>%
  vline_right(border=officer::fp_border(color="black")) %>%
  fix_border_issues()
```

	Obesity	
Low Educational Level	No	Yes
No	492 (74.7%)	167 (25.3%)
Yes	206 (55.5%)	165 (44.5%)

```
tIO <- table(dt$Low.Educational.Level,2-dt$Obesity)
chisq.test(tIO)
```

```
    Pearson's Chi-squared test with Yates' continuity correction

data:  tIO
X-squared = 38.909, df = 1, p-value = 4.44e-10
```

```
summary(epi2x2(tIO))
```

```
Epidemiological 2x2 Table Analysis

Input Matrix:

      1   2
  0 167 492
  1 165 206

Pearson Chi-Squared Statistic (Includes Yates' Continuity Correction): NA
Associated p.value for H0: There is no association between exposure and outcome vs. HA: There is a
p.value using Fisher's Exact Test (1 DF) : 0

Estimate of Odds Ratio: 0.424
95% Confidence Limits for true Odds Ratio are: [0.324, 0.555]

Estimate of Relative Risk (Cohort, Col1): 0.57
95% Confidence Limits for true Relative Risk are: [0.479, 0.678]
```

```
Estimate of Risk Difference (p1 - p2) in Cohort Studies: -0.191
95% Confidence Limits for Risk Difference: [-0.254, -0.129]

Estimate of Risk Difference (p1 - p2) in Case Control Studies: -0.202
95% Confidence Limits for Risk Difference: [-0.264, -0.14]

Note: Above Confidence Intervals employ a continuity correction.
```

Now we fit (by maximum likelihood) a logistic regression model for the relationship between Obesity and low educational level adjusting by age:

```
ml <- glm(Obesity ~ Age + Low.Educational.Level,family=binomial(link=logit),data=dt)
summary(ml)
```

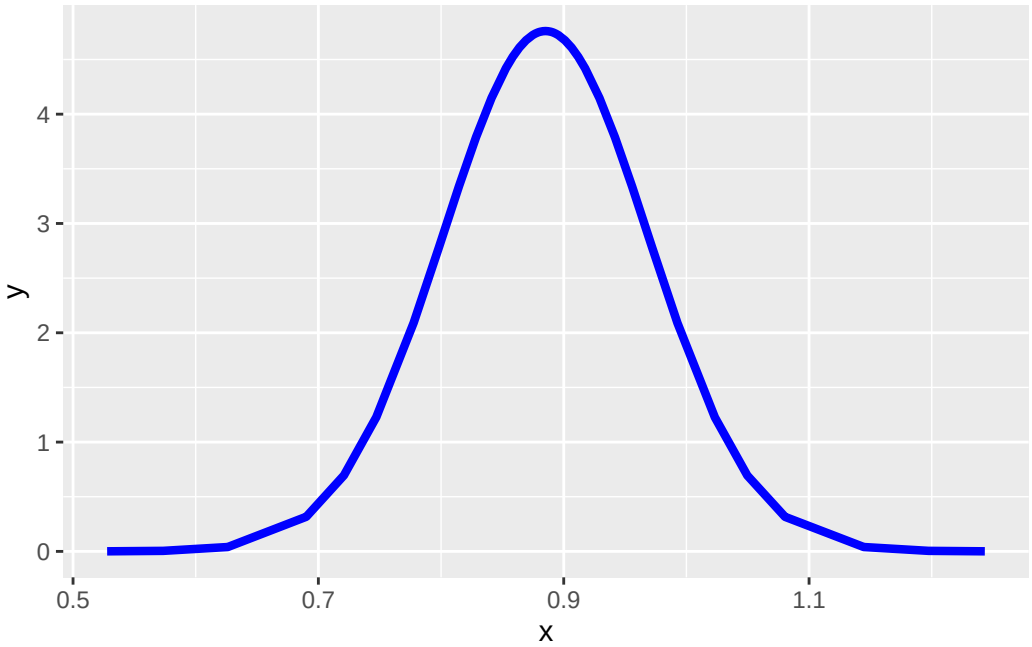
```
Call:
glm(formula = Obesity ~ Age + Low.Educational.Level, family = binomial(link = logit),
    data = dt)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3261  -0.8626  -0.7157   1.2361   1.8066

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)          -2.10003    0.29693  -7.073 1.52e-12 ***
Age                   0.02286    0.00627   3.646 0.000266 ***
Low.Educational.Level  0.61433    0.15300   4.015 5.94e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1294.9  on 1029  degrees of freedom
Residual deviance: 1242.5  on 1027  degrees of freedom
AIC: 1248.5

Number of Fisher Scoring iterations: 4
```

The bayesian estimation is:

```
formula <- Obesity ~ 1 + Age + Low.Educational.Level
m.log <- inla(formula,family="binomial",data=dt,
              control.fixed=list(mean=list(Age=0, Low.Educational.Level=1),
                                 prec=list(Age=0.0001, Low.Educational.Level=100),
                                 mean.intercept=0, prec.intercept=0.000001),
              control.predictor=list(compute=TRUE),control.compute=list(dic=TRUE, cpo=TRUE))
smry <- summary(m.log)
smry$fixed <- smry$fixed[,1:5]
smry
```

```
Fixed effects:
                        mean    sd 0.025quant 0.5quant 0.975quant
(Intercept)           -1.986 0.293     -2.567   -1.984     -1.415
Age                    0.018 0.006      0.007    0.018      0.030
Low.Educational.Level  0.885 0.084      0.721    0.885      1.050

Deviance Information Criterion (DIC) ...............: 1250.24
Deviance Information Criterion (DIC, saturated) ....: -7082.37
Effective number of parameters .....................: 2.31

 is computed
```

```
dpost.E <- m.log$marginals.fixed$Low.Educational.Level

ggplot(data.frame(x=dpost.E[,1],y=dpost.E[,2]),aes(x,y))+
  geom_line(col="blue",size=1.5)
```



4. Tobacco-attributable cancer mortality in the Canary Islands

Data reading:

```
dtab <- read_excel("data/M_tabaco.xlsx")
if (is_html_output()){
  dtab %>%
  kbl() %>%
  kable_styling(c("striped", "hover"),full_width = FALSE) %>%
  scroll_box(height = "250px")
} else{
  head(dtab) %>% flextable() %>% autofit()
}
```

Year	G.Age	Sex.Male	N	D_01	D_02	D_03	D_04	D_05	D_06
1,980	1	1	83,202	1	2	0	3	7	0
1,980	1	0	83,202	0	1	0	0	1	0
1,980	2	1	75,904	6	4	1	5	33	2
1,980	2	0	78,823	0	0	0	0	4	0
1,980	3	1	48,170	9	11	6	6	53	1
1,980	3	0	54,008	0	0	0	2	4	1

Each row in the data file corresponds to the number of deaths due to different types of cancer in the Canary Islands according to year (from 1980 to 2002), sex and age group (4 age groups are considered). In particular, we will analyse the number of deaths from lung cancer attributable to smoking (variable `D_05`) among subjects in age group 2 (between 45 and 55 years). To do so, we will fit a Poisson model, considering as offset the number of deceased subjects in the population for each year, sex and age group. Frequentist estimation using maximum likelihood produces the following result:

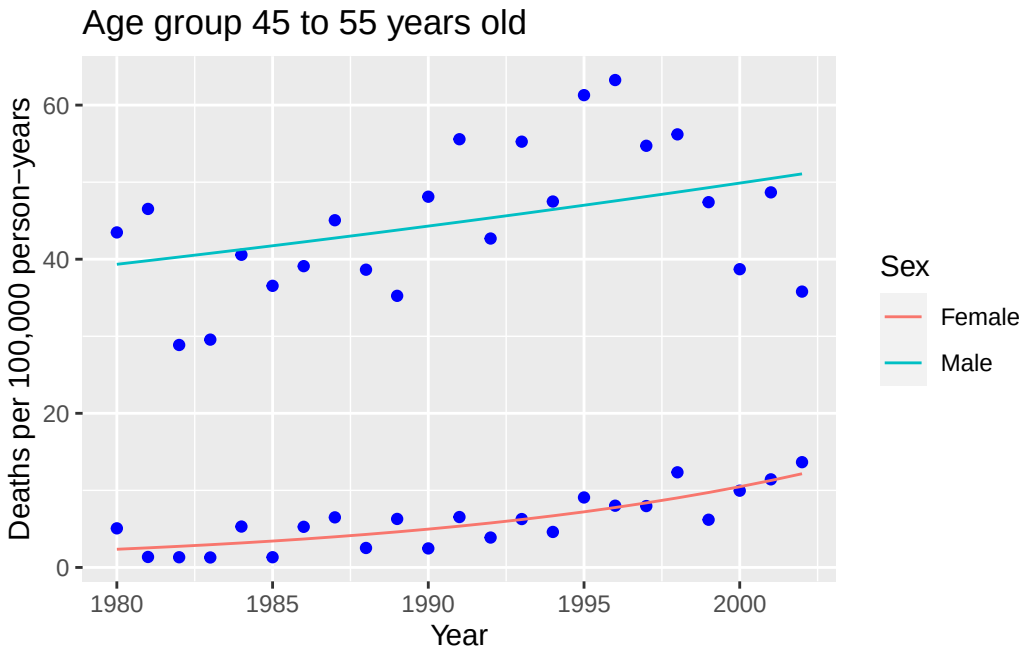
```
dtab$Sex <- ifelse(dtab$Sex.Male==1,"Male","Female")
dtabg2 <- dtab %>%
  filter(G.Age==2) %>%
  mutate(Rate=100000*D_05/N)

mtab <- glm(D_05 ~ offset(log(N)) + Sex.Male*Year, family=poisson, data=dtabg2)
summary(mtab)
```

```
Call:
glm(formula = D_05 ~ offset(log(N)) + Sex.Male * Year, family = poisson,
    data = dtabg2)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.41941  -0.78271  -0.03452   0.67019   2.03603

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   -158.23354   29.53446  -5.358 8.43e-08 ***
```

```
Sex.Male       126.89682   31.19693   4.068 4.75e-05 ***
Year             0.07453    0.01481   5.034 4.80e-07 ***
Sex.Male:Year   -0.06267    0.01564  -4.007 6.16e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 720.508  on 45  degrees of freedom
Residual deviance:  45.066  on 42  degrees of freedom
AIC: 254.11

Number of Fisher Scoring iterations: 4
```

The following figure shows the fit of the model:

```
b <- mtab$coeff
pd <- as.numeric(predict(mtab))
dtabg2 <- dtabg2 %>%
  mutate(Adj.Rate = 100000*exp(pd)/N)

ggplot(dtabg2,aes(Year,Rate))+
  geom_point(colour="blue")+
  geom_line(aes(Year,Adj.Rate,colour=Sex))+
  labs(x="Year",y="Deaths per 100,000 person-years",
       title="Age group 45 to 55 years old")
```



Now the bayesian estimation of the same model:

```
### inla
formula <- D_05 ~ offset(log(N)) + Sex.Male*Year
m.pois <- inla(formula,family="poisson",data=dtabg2,
               control.fixed=list(mean=list(Sex.Male=0, Year=0, `Sex.Male:Year`=0),
                                  prec=list(Sex.Male=0.000001, Year=0.000001, `Sex.Male:Year`=0.00
                                  mean.intercept=0, prec.intercept=0.000001),
              control.predictor=list(compute=TRUE),control.compute=list(dic=TRUE, cpo=TRUE))
smry <- summary(m.pois)
smry$fixed <- smry$fixed[,1:5]
smry
```

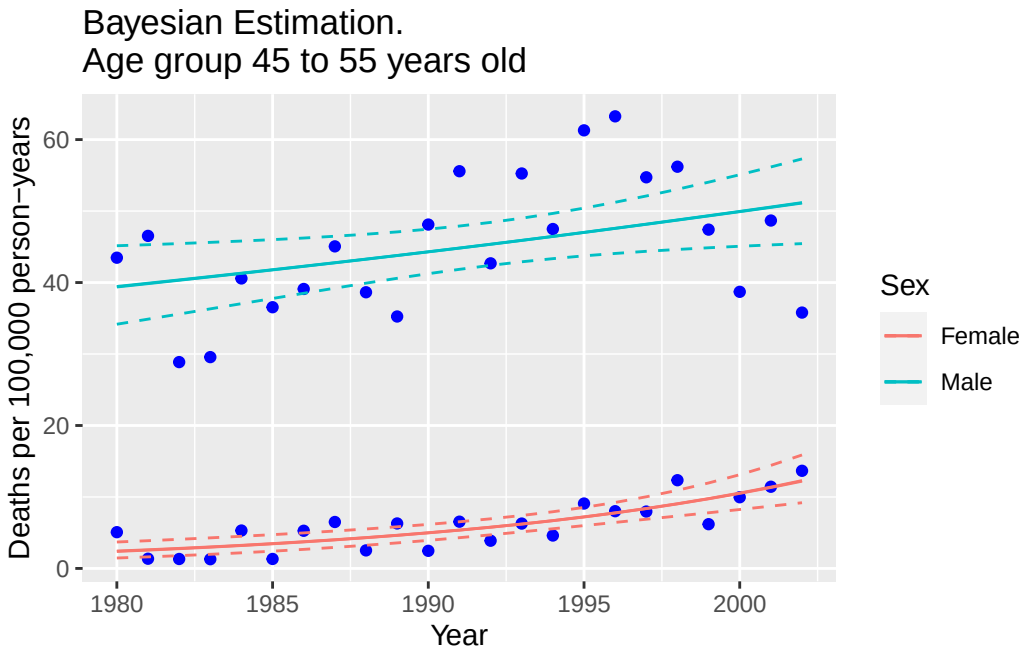
```
Fixed effects:
                  mean     sd 0.025quant 0.5quant 0.975quant
(Intercept)   -157.984 29.501   -216.986 -157.602   -101.208
Sex.Male       126.698 31.163     66.531  126.351    188.834
Year             0.074  0.015      0.046    0.074      0.104
Sex.Male:Year   -0.063  0.016     -0.094   -0.062     -0.032

Deviance Information Criterion (DIC) ...............: 254.19
Deviance Information Criterion (DIC, saturated) ....: -558.17
```

```
Effective number of parameters ......................: 4.02

 is computed
```

```
#m.pois$all.hyper$fixed
```

The following figure shows the bayesian credibility intervals for the predictions of the model:

```
fitBayes <- m.pois$summary.fitted.values
dtabg2 <- dtabg2 %>%
  mutate(`Adj.Rate (Bayes)` = 100000*fitBayes$mean/N,
         adj.median=100000*fitBayes[["0.5quant"]]/N,
         adj.q2.5=100000*fitBayes[["0.025quant"]]/N,
         adj.q97.5=100000*fitBayes[["0.975quant"]]/N)

ggplot(dtabg2,aes(Year,Rate))+
  geom_point(colour="blue")+
  geom_line(aes(Year,`Adj.Rate (Bayes)`,colour=Sex))+
    geom_line(aes(Year,adj.median,colour=Sex))+
    geom_line(aes(Year,adj.q2.5,colour=Sex),linetype=2)+
    geom_line(aes(Year,adj.q97.5,colour=Sex),linetype=2)+
  labs(x="Year",y="Deaths per 100,000 person-years",
       title="Bayesian Estimation.\nAge group 45 to 55 years old")
```



Alternatively we could have modelled these data using a negative binomial model, but the fit does not improve substantially (it is even slightly worse). The estimated parameter values are almost unchanged:

```
### inla
m.nbin <- inla(formula,family="nbinomial",data=dtabg2,
               control.fixed=list(mean=list(Sex.Male=0, Year=0, `Sex.Male:Year`=0),
                                  prec=list(Sex.Male=0.000001, Year=0.000001, `Sex.Male:Year`=0.00
                                  mean.intercept=0, prec.intercept=0.000001),
            control.predictor=list(compute=TRUE),control.compute=list(dic=TRUE, cpo=TRUE))
smry <- summary(m.nbin)
smry$fixed <- smry$fixed[,1:5]
smry
```

```
Fixed effects:
                  mean     sd 0.025quant 0.5quant 0.975quant
(Intercept)   -157.761 30.236   -218.235 -157.358    -99.579
Sex.Male       125.050 32.496     62.141  124.742    189.703
Year             0.074  0.015      0.045    0.074      0.105
Sex.Male:Year   -0.062  0.016     -0.094   -0.062     -0.030
```

```
Model hyperparameters:
                                                              mean        sd
size for the nbinomial observations (1/overdispersion) 26063.09 405585.09
                                                         0.025quant 0.5quant
size for the nbinomial observations (1/overdispersion)      30.56   160.36
                                                         0.975quant mode
size for the nbinomial observations (1/overdispersion)   49390.00   NA

Deviance Information Criterion (DIC) ...............: 254.47
Deviance Information Criterion (DIC, saturated) ....: -557.89
Effective number of parameters .....................: 4.58

 is computed
```

5. Mixed-models.

We consider a sample of n subjects from a given population. In the i-th subject and for a set of covariates $x_{i,1},...,x_{i,n_i}$ we observe the corresponding random variables $y_{i,1},...,y_{i,n_i}$, which obey the mixed model:

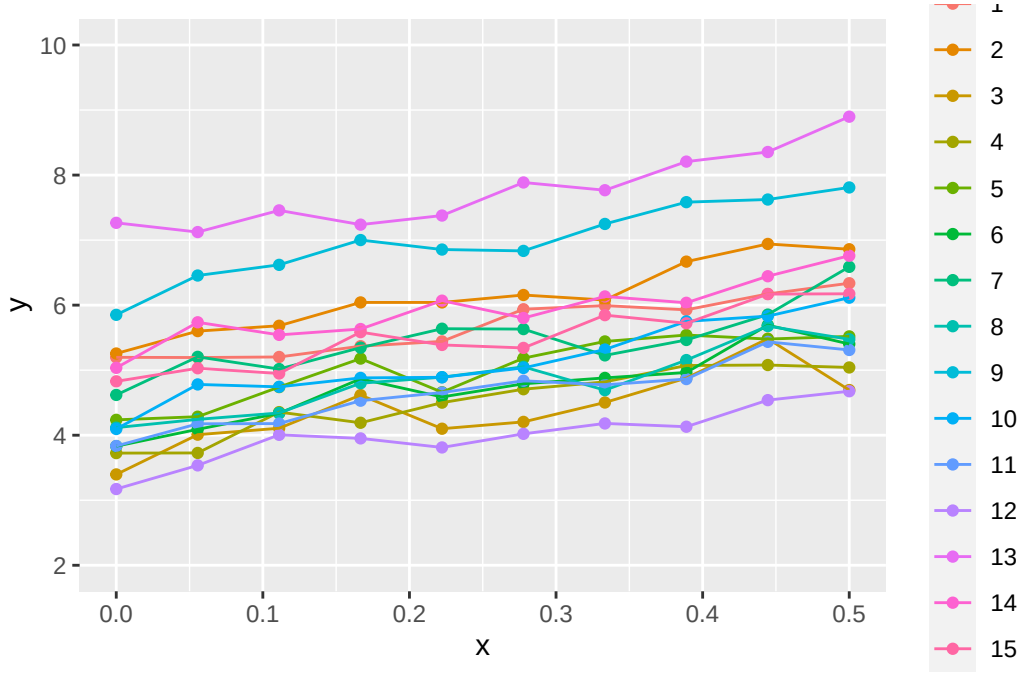
$$y_{i,j} = \beta_0 + \beta_1.x_{i,j} + a_i + e_{i,j}$$

Here, $a_1,...,a_n$ are iid $N(0,\sigma_a)$ random variables which denote the random effects of individuals and for each $i = 1,...,n$, $e_{i,1},...,e_{i,n_i}$ are iid random variables $N(0,\sigma_e)$ that correspond to within-subject variability .

Mixed-models 1. We begin by simulating this model:

```
set.seed(19941998)
n=15    ### Number of subjects
m=10    ### Number of observations per subject
b <- c(5,3)
a <- rnorm(n,0,1)
z <- seq(0,0.5,length=m)
```

```
dt <- data.frame(id=gl(n,m),x=rep(z,n)) %>%
  mutate(y=b[1]+b[2]*x+a[as.integer(id)]+rnorm(n*m,0,0.2))

ggplot(dt,aes(x,y,color=id))+
  geom_point()+
  geom_line(aes(group=id))+
  ylim(c(2,10))
```



The frequentist estimation is:

```
lmm <- lmer(y ~ x + (1 | id), data = dt,REML = FALSE)
summary(lmm)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: y ~ x + (1 | id)
   Data: dt

     AIC      BIC   logLik deviance df.resid
    39.0     51.0    -15.5     31.0      146

Scaled residuals:
```

```
    Min      1Q  Median      3Q     Max
-2.3126 -0.5656 -0.1156  0.5606  2.5212

Random effects:
 Groups   Name        Variance Std.Dev.
 id       (Intercept) 0.94190  0.9705
 Residual             0.04187  0.2046
Number of obs: 150, groups:  id, 15

Fixed effects:
            Estimate Std. Error t value
(Intercept)   4.6602     0.2525   18.46
x             2.9008     0.1047   27.71

Correlation of Fixed Effects:
  (Intr)
x -0.104
```

And the bayesian estimation:

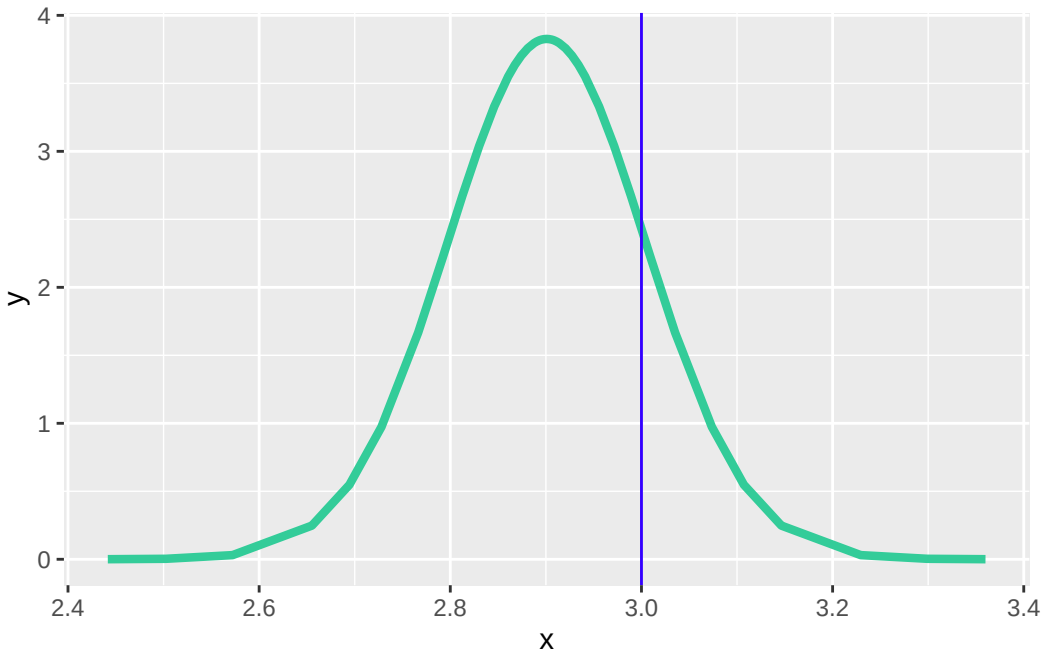
```
formula <- y ~ 1 + x + f(id,model="iid")
bmm <- inla(formula,family="gaussian",data=dt,
           control.fixed=list(mean=0, prec=0.000001,
           mean.intercept=0, prec.intercept=0.000001))
smry <- summary(bmm)
smry$fixed <- smry$fixed[,1:5]
smry
```

```
Fixed effects:
             mean    sd 0.025quant 0.5quant 0.975quant
(Intercept) 4.660 0.261      4.143    4.660      5.177
x           2.901 0.105      2.694    2.901      3.107

Model hyperparameters:
                                          mean    sd 0.025quant 0.5quant
Precision for the Gaussian observations 24.05 2.918     18.726    23.90
Precision for id                         1.13 0.401      0.509     1.07
                                        0.975quant mode
Precision for the Gaussian observations      30.21   NA
```

```
Precision for id                                2.07    NA

 is computed
```

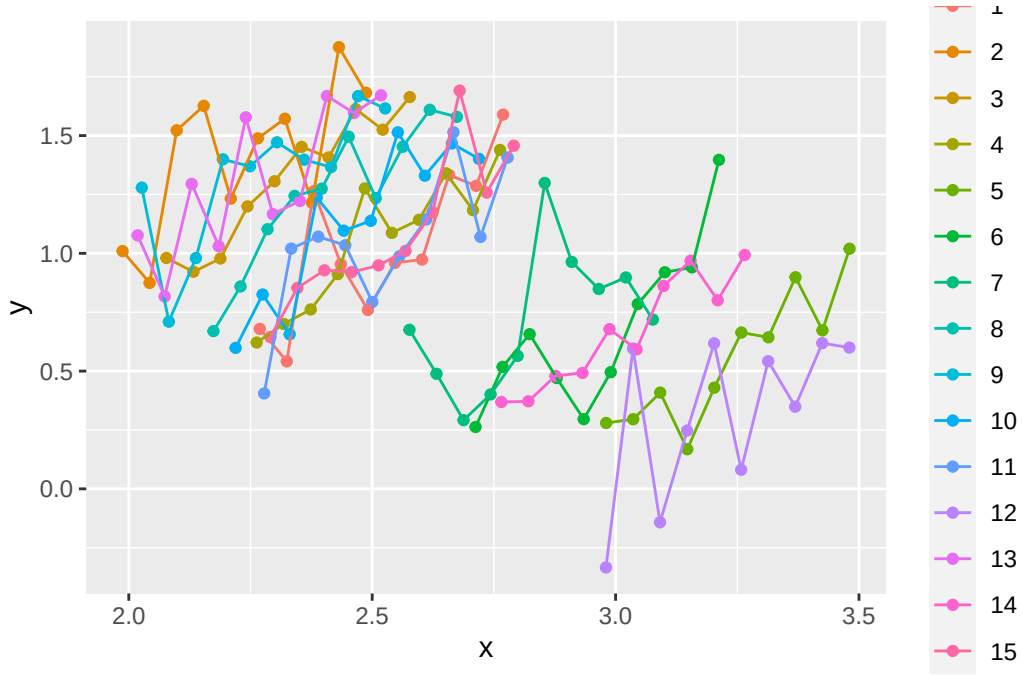
Posterior density.

```
### Densidad a posteriori
dpost.intercept <- data.frame(bmm$marginals.fixed$`(Intercept)`)
dpost.x <- data.frame(bmm$marginals.fixed$x)

ggplot(dpost.x,aes(x,y))+
  geom_line(colour="#33CC99",linewidth=1.5)+
  geom_vline(xintercept=3,colour="#3300FF")
```



Mixed-models 2. Now we simulate another dataset in which each individual has its own slope, different from the general slope for the whole population.

```
n=15     ### Number of subjects
m=10     ### Number of observations per subject
a <- rnorm(n,2.5,0.3)
b <- 3-a
z <- rep(seq(0,0.5,length=m),n)
dt <- data.frame(id=gl(n,m)) %>%
  mutate(x=z+a[as.integer(id)],
         y=b[as.integer(id)]+1.5*z+rnorm(n*m,0,0.2))

ggplot(dt,aes(x,y,color=id))+
  geom_point()+
  geom_line(aes(group=id))
```



Frequentist Estimation:

- Simple linear model (bad idea)

```
ml <- lm(y ~ x, data=dt)
summary(ml)
```

```
Call:
lm(formula = y ~ x, data = dt)

Residuals:
     Min       1Q   Median       3Q      Max
-1.14726 -0.30964  0.01695  0.31673  0.78883

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.29653    0.23184   9.905  < 2e-16 ***
x           -0.49756    0.08749  -5.687 6.68e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3927 on 148 degrees of freedom
Multiple R-squared:  0.1793,    Adjusted R-squared:  0.1738
F-statistic: 32.34 on 1 and 148 DF,  p-value: 6.678e-08
```

- Linear mixed model

```
lmm <- lmer(y ~ x + (1 | id), data = dt,REML = FALSE)
summary(lmm)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: y ~ x + (1 | id)
   Data: dt

     AIC      BIC   logLik deviance df.resid
     5.3     17.3      1.4     -2.7      146

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.05300 -0.60358 -0.04049  0.53140  2.92777

Random effects:
 Groups   Name        Variance Std.Dev.
 id       (Intercept) 0.59183  0.7693
 Residual             0.03434  0.1853
Number of obs: 150, groups:  id, 15
```

```
Fixed effects:
            Estimate Std. Error t value
(Intercept) -2.63440    0.31643  -8.325
x            1.38120    0.09367  14.745

Correlation of Fixed Effects:
  (Intr)
x -0.777
```

Bayesian Estimation.

```
### Bayesian estimation
formula <- y ~ 1 + x + f(id,model="iid")
bmm <- inla(formula,family="gaussian",data=dt,
           control.fixed=list(mean=0, prec=0.001,
           mean.intercept=0, prec.intercept=0.001))
smry <- summary(bmm)
smry$fixed <- smry$fixed[,1:5]
smry
```

```
Fixed effects:
              mean    sd 0.025quant 0.5quant 0.975quant
(Intercept) -2.623 0.327     -3.273   -2.619     -1.989
x            1.377 0.097      1.186    1.377      1.565

Model hyperparameters:
                                          mean   sd 0.025quant 0.5quant
Precision for the Gaussian observations 29.32 3.56     22.807    29.16
Precision for id                         1.80 0.67      0.801     1.70
                                        0.975quant mode
Precision for the Gaussian observations      36.83   NA
Precision for id                              3.40   NA

 is computed
```

6. Nonlinear regression: autoregresive processes.

National Morbidity, Mortality, and Air Pollution Study. The National Morbidity, Mortality and Air Pollution Study (NMMAPS) is a large time series study designed to estimate the effect of air pollution on the health of individuals living in 108 US cities during the period 1987-2000. Data on the daily concentration of particulate matter with an aerodynamic diameter less than 10 (PM_{10}) and nitrogen dioxide (NO_2), both measured in $\mu g/m^3$, as well as the daily temperature for Salt Lake City are contained in the `NMMAPSraw.csv` file. We use this data set to study the relationship between PM_{10} and temperature as an illustration of a linear regression model.

We denote by $PM_{T,M,d}$ the concentration of particles with aerodynamic diameter less than 10 μm according to temperature (T) month (M) and day (d). We assume that $PM_{T,M,d} \sim N\left(\mu_{T,M,d}, \sigma\right)$, where:

$$\mu_{T,M,d} = \beta_0 + \beta_1 \cdot T + \gamma_M + z_d$$

γ_M denotes the effect of month M (January is taken as the reference month, hence γ_{Jan} =0). We also assume that z_d, $d = 1, ..., N$ is an autoregressive process of order 1 and hence, $z_d \sim N\left(\phi \cdot z_{d-1}, \sigma\right)$.

Data reading:

```
dataNMMAPS <- read_excel("data/NMMAPSraw.xlsx")
if (is_html_output()){
  dataNMMAPS %>%
  kbl() %>%
  kable_styling(c("striped", "hover"),full_width = FALSE) %>%
  scroll_box(height = "250px")
} else{
  head(dataNMMAPS) %>% flextable() %>% autofit()
}
```

date	pm10no2	temperature	id
01 January 1987	63.33052641.214043660000002	29.5	1
02 January 1987	15.07634130.56186975	34.0	2
03 January 1987	15.53401028.714043660000002	37.0	3
04 January 1987	17.15268110.380710329999999	42.5	4
05 January 1987	7.83052627.08360888	35.0	5
06 January 1987	23.15268143.297376989999997	29.5	6

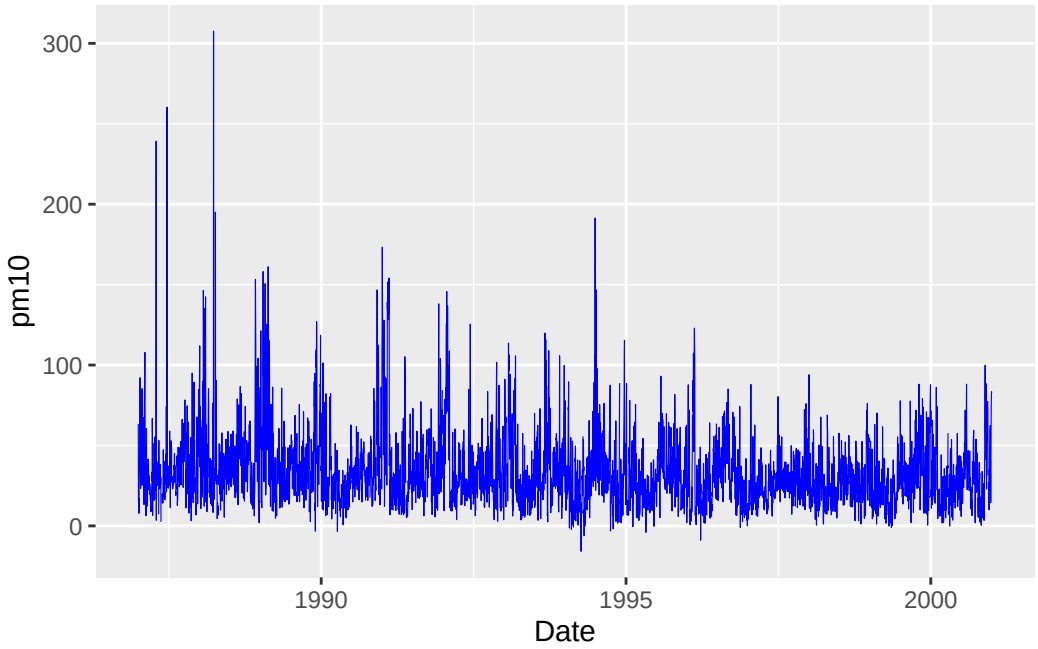
The total number of observations in this file is:

```
nrow(dataNMMAPS)
```

```
[1] 5114
```

Next we represent the concentration of `pm10` vs. date:

```
dataNMMAPS <- dataNMMAPS %>%
  mutate(date=dmy(date,locale="en_US.UTF-8"),
         month=factor(month(date,locale="en_US.UTF-8",
                            label=TRUE,abbr=TRUE),ordered=FALSE))

ggplot(dataNMMAPS,aes(date,pm10))+
  geom_line(cex=0.2,color="blue")+labs(x="Date")
```



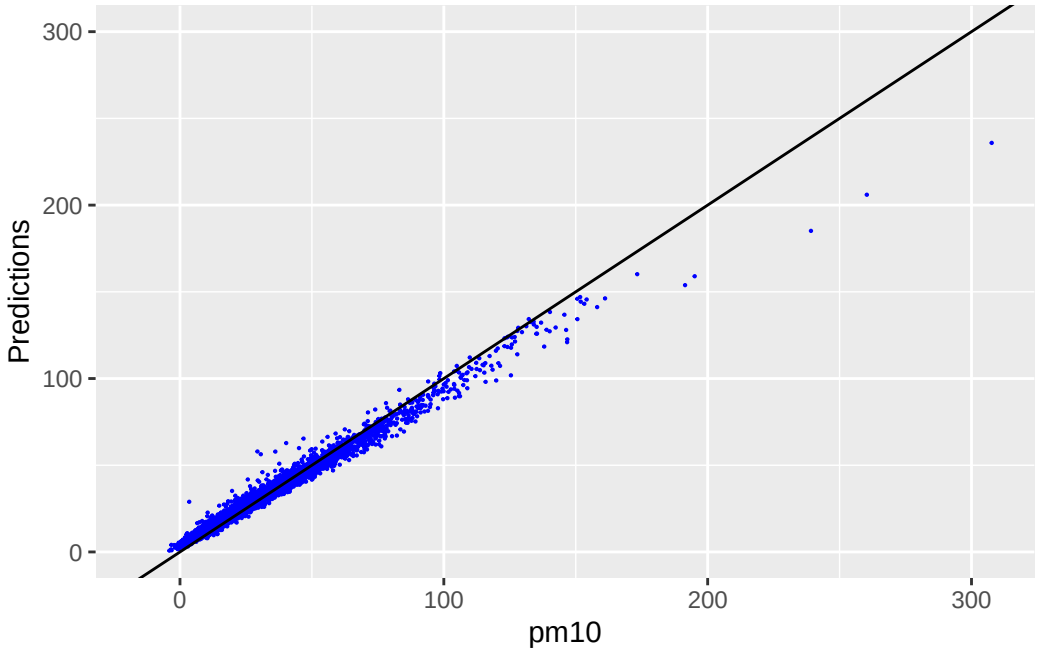
Bayesian estimation:

```
### Process
formula <- pm10 ~ 1 + temperature + month +
  f(id, model="ar1",hyper = list(prec = list(prior="loggamma",param=c(1,0.01))))

ml <- inla(formula,family="gaussian",data=dataNMMAPS,control.predictor = list(compute = TRUE))
round(ml$summary.fixed[,1:5],3)
```

```
              mean    sd 0.025quant 0.5quant 0.975quant
(Intercept)  29.287 2.591     24.185   29.294     34.350
temperature   0.413 0.047      0.321    0.413      0.504
monthFeb     -5.966 2.783    -11.429   -5.965     -0.510
monthMar    -19.245 2.980    -25.079  -19.249    -13.388
monthApr    -26.528 3.103    -32.614  -26.529    -20.438
monthMay    -28.549 3.223    -34.866  -28.552    -22.220
monthJun    -26.070 3.425    -32.787  -26.070    -19.351
monthJul    -27.577 3.592    -34.632  -27.573    -20.543
monthAug    -24.525 3.549    -31.488  -24.524    -17.566
monthSep    -21.748 3.349    -28.322  -21.746    -15.184
monthOct    -17.879 3.114    -23.987  -17.879    -11.771
monthNov    -12.772 2.968    -18.588  -12.775     -6.943
monthDec      0.320 2.757     -5.083    0.319      5.734
```

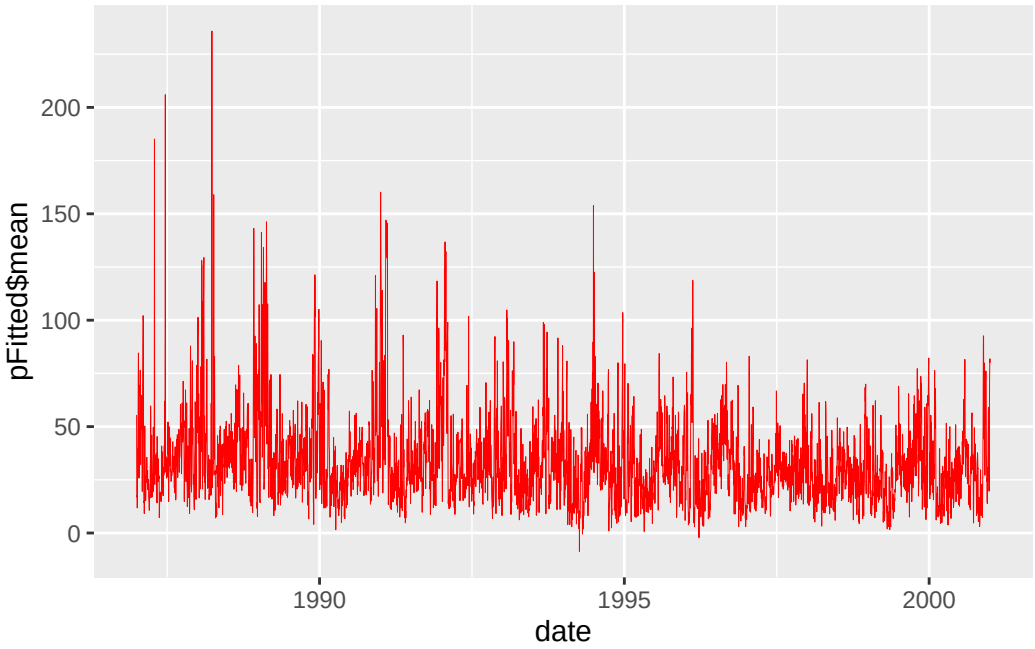
The following figure shows the predicted vs. observed values of `pm10`:

```
pFitted <- ml$summary.fitted.values
ggplot(data.frame(pm10=dataNMMAPS$pm10,fittedMean=pFitted$mean),
       aes(pm10,fittedMean))+
  geom_point(size=.1,col="blue")+
  geom_abline(intercept=0,slope=1)+ylim(c(0,300))+
  labs(y="Predictions")
```



Fitted Values:

```
ggplot(data.frame(date=dataNMMAPS$date,pFitted$mean),aes(date,pFitted$mean))+
  geom_line(col="red",size=0.1)
```



7. INLA on the web

Here is a list of some free resources on the web to continue learning Bayesian methods with INLA:

- Home page of R-INLA Project
- Bayesian inference with INLA. Virgilio Gómez Rubio
- Geospatial Health Data: Modeling and Visualization with R-INLA and Shiny. Paula Moraga
- INLA examples & tutorials
- INLA for (generalized) linear models
- Bayesian Regression Modeling with INLA, Xiaofeng Wang, Yu Ryan Yue, Julian Faraway github site. Examples

Some interesting papers on INLA:

- New Frontiers in Bayesian Modeling Using the INLA Package in R Van Niekerk, J., Bakka, H., Rue, H., & Schenk, O. *Journal of Statistical Software*, 100, 1–28. (2021).

- Spatial Data Analysis with R-INLA with Some Extensions. Bivand RS, Gómez-Rubio V, Rue H. *Journal of Statistical Software* 63, 20 (2015)
- Bayesian Computing with INLA: A Review Rue, H. Riebler A, Sørbye SH, Illian JB, Simpson DP. *Annu. Rev. Stat. Appl.* 4:395–421 (2017)