Bayesian Statistics

Class Assignment 1

Simon Roth
nomis.roth@gmx.net

28.11.2018

Packages

Note: This is powerful R-markdown document which combines the entire code, outputs and text. If you don't have pacman installed, just do it once and it will manage all the rest of the dependecies.

```
pacman::p_load(rjags, dplyr, purrr, tidyr, ggplot2, broom, rjags, texreg, ggthemes, janitor, knitr)
ggplot2::theme_set(theme_bw())
set.seed(2018)
```

Data

A reduced dataset of Student Panel Survey during the Lecture in Introduction to Political Methodology Winter term 2016/2017 at the University of Konstanz

- `poleff` Political Efficacy (Likert Score based on 7 items) A larger value = higher level of efficacy
- `friend` Number of alteri in friendship network
- `poldisc` Number of alteri in political discussion network
- `lr.self` Ideological orientation (left right self-placement) 1: Left <- -> 11: Right
- `lr.self.2` Ideological orientation (left right self-placement, second measurement) 1: Left <- -> 11: Right
- `univ.election` Vote intention at the next university election. 1: Yes; 0: other (No and DK)
- `polint` interest at university politics 1: not interested at all <- -> 5 strongly interested
- `tuition` opinion on the general tuition fee for German universities 1: support; 2: reject; 3: indifferent
- `acceptable` acceptable level of the tuition fee (in Euro per Semester) (Only those who support the tuition fee or indifferent)
- protest1 - protest6 willingness to participate a protest action against the general tuition fee 1: yes; 0: no
  - `protest1` demonstration in Konstanz
  - `protest2` demonstration in Stuttgart
  - `protest3` giving signature at petitions
  - `protest4` strike
  - `protest5` occupation of university buildings
  - `protest6` legal dispute at courts

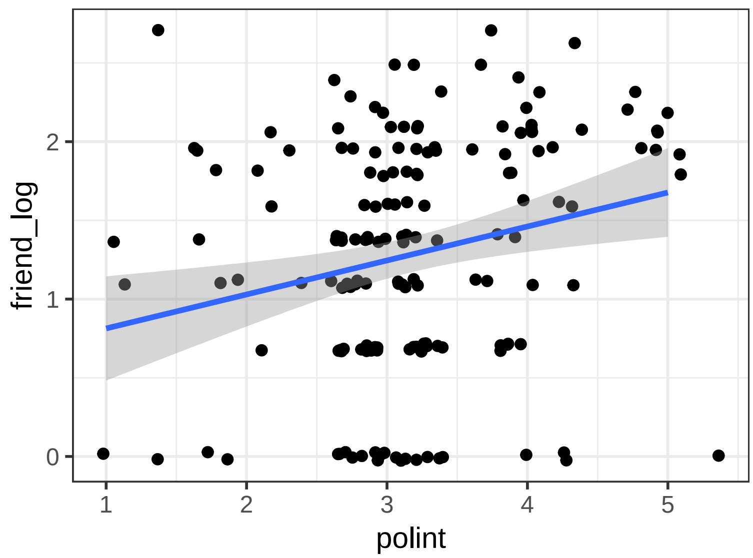
```
dat <- get(load("../data/Bayes_Student_Survey.RData")) %>%
  mutate(friend_log = log(friend + 1))

# glimpse(dat)
```

1 Frequentist Estimation

Estimate the parameters of a bivariate regression via OLS. You can choose a dependent variable and one independent variable from the dataset for yourself.

Political interest might influence the general openess to communicate and make friends.

```
dat %>%
  ggplot(aes(polint, friend_log)) +
  geom_jitter() +
  geom_smooth(method = "lm")
```




```
lm(friend_log ~ polint, data = dat) %>%
  texreg::texreg(float.pos = "ht!")
```

	Model 1
(Intercept)	0.60^{*}
	(0.24)
polint	0.22^{**}
	(0.07)
R^2	0.05
Adj. R^2	0.05
Num. obs.	165
RMSE	0.73

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 1: Statistical models

2 Bayesian Estimation

Run the MCMC to obtain the posterior of the same regression model above with 5 chains. You have to run the first 200 iterations without collecting posterior. Thereafter collect your posterior in 1000 iterations. Use the same prior as on the slides.

```r
reg.model <- "model{
  for (i in 1:N){
    y[i] ~ dnorm(mu[i], tau)
    mu[i] <- beta0 + beta1 * x[i]
  }

  beta0 ~ dnorm(0, 0.0001)
  beta1 ~ dnorm(0, 0.0001)

  tau ~ dgamma(0.001, 0.001)
  sigma <- 1/sqrt(tau)
}"

write(reg.model, "Bayes_Bivariate_Reg_Student_Survey.bug")
```

```r
jags.data <- list(
  y = dat$friend_log,
  x = dat$polint,
  N = nrow(dat)
)

jags.inits <- 1:5 %>%
  map(~ list(beta1 = runif(1, min = -100, max = 100) %>% round()))

jags.reg <- jags.model(
  file = "Bayes_Bivariate_Reg_Student_Survey.bug",
  inits = jags.inits,
  data = jags.data,
  n.chains = length(jags.inits)
)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 165
##    Unobserved stochastic nodes: 3
##    Total graph size: 350
##
## Initializing model
```

```r
update(jags.reg, 200)

jags.reg.out <- coda.samples(
  jags.reg,
  variable.names = c("beta0", "beta1", "sigma"),
  n.iter = 1000,
  thin = 1
)

jags.reg.out %>%
  summary() %>%
```

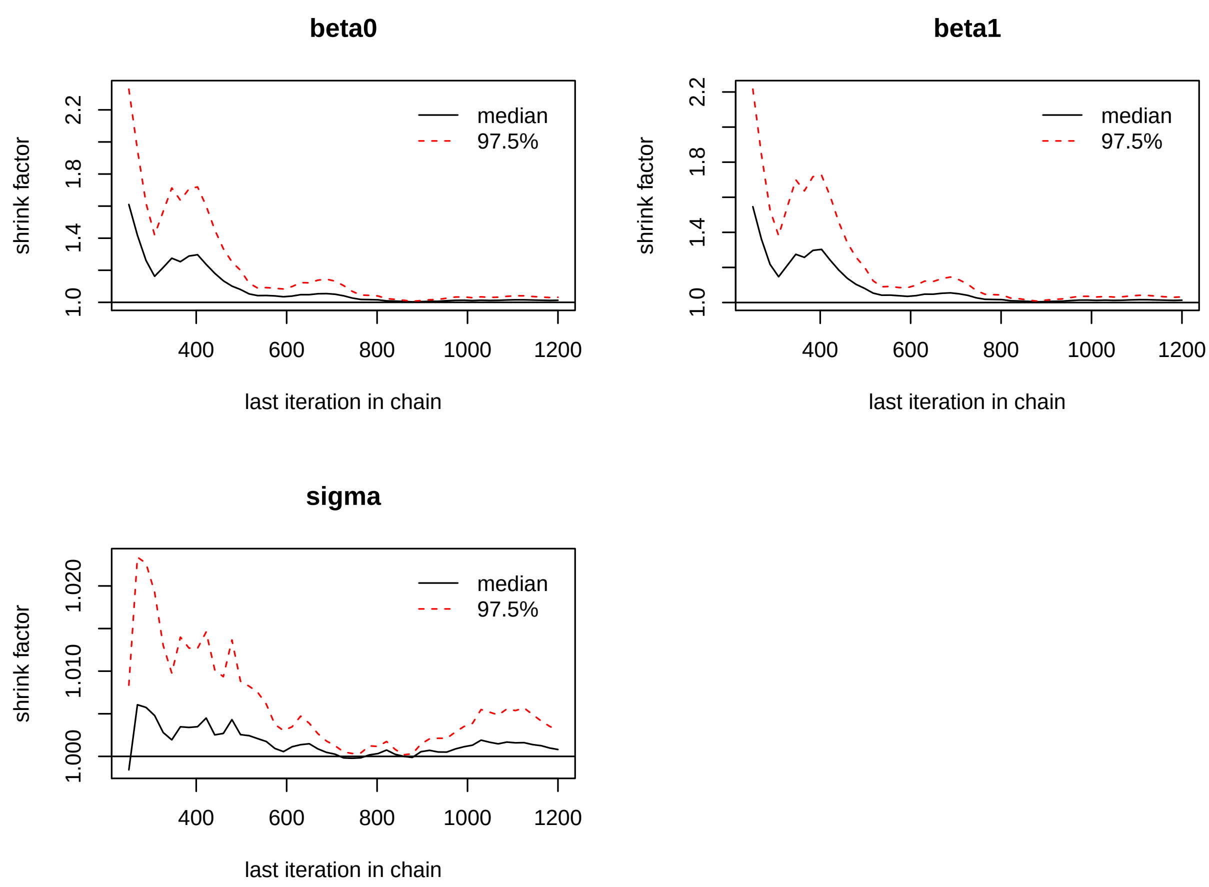
```
.$statistics %>%
kable
```

	Mean	SD	Naive SE	Time-series SE
beta0	0.6036442	0.2385909	0.0033742	0.0194307
beta1	0.2143240	0.0728827	0.0010307	0.0059053
sigma	0.7363239	0.0414159	0.0005857	0.0005913

3 Check Cnvergence

based on visible inspection and the Gelman- Rubin-Statistics.

```
gelman.plot(jags.reg.out)
```

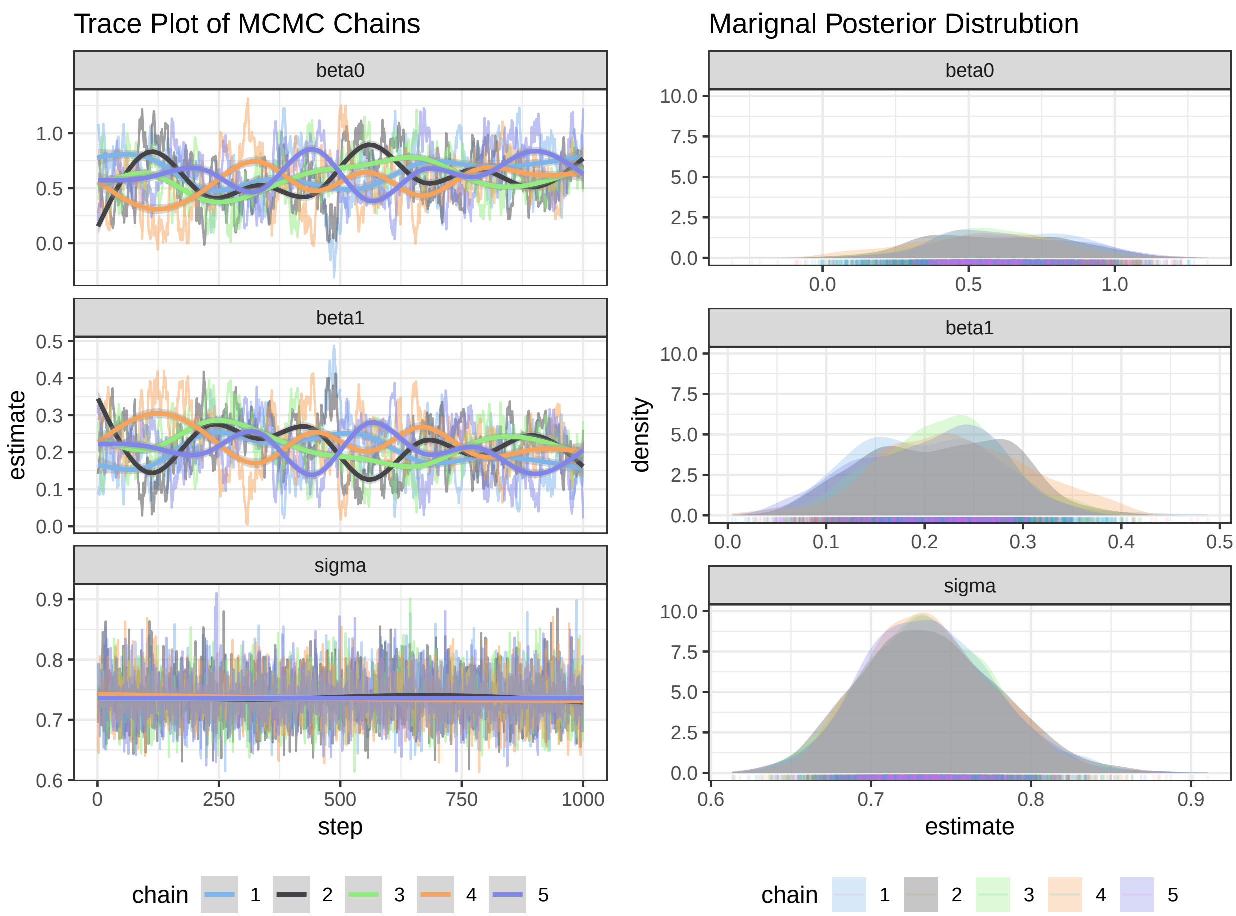



```
#autocorr.plot(jags.reg.out)
```

4 Report the posterior

by using summary() and plot().

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



5 Calculate the probability that beta1 is positive

Which percentage of posterior is greater than zero (positive)?

```
unlist(jags.reg.out) %>%
  tibble(p = . > 0) %>%
  tabyl(p) %>%
  kable
```

p	n	percent
FALSE	19	0.0012667
TRUE	14981	0.9987333

Answer: 99%

6 Repeat 2-4 with different prior.