Applied Bayesian Statistics

Class Assignment 1

Simon Roth
nomis.roth@gmx.net

5.12.2018

Packages

Note *This is Rmarkdown 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 (for ever).*

```
# install.packages("pacman")
pacman::p_load(
  rjags, dplyr, purrr, tidyr, ggplot2, broom,
  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

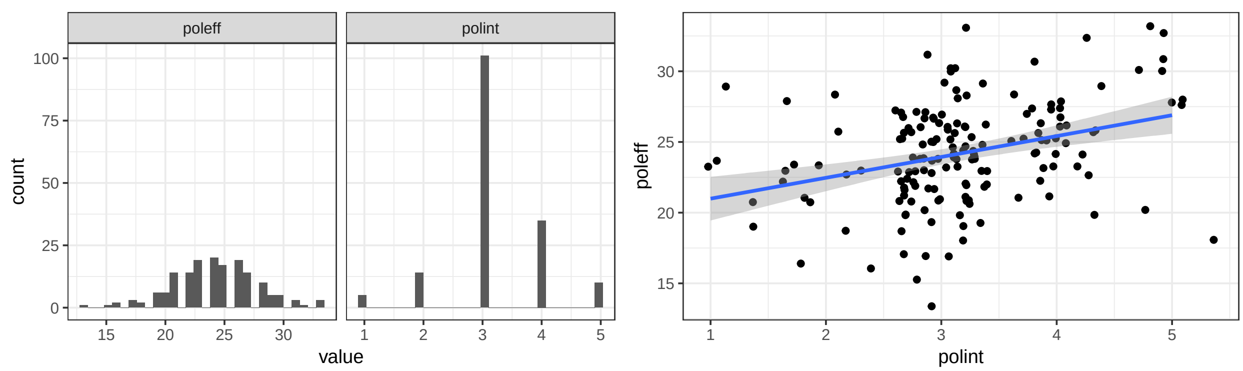
```
dat <- get(load("data/Bayes_Student_Survey.RData"))
  #mutate(friend_log = log(friend + 1))
```

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.

```
left_side <- dat %>%
  select(polint, poleff) %>%
  gather(var, value) %>%
  ggplot(aes(value)) +
  geom_histogram() +
  facet_wrap( ~ var, scales = "free_x")

right_side <- dat %>%
  ggplot(aes(polint, poleff)) +
  geom_jitter() +
  geom_smooth(method = "lm")

gridExtra::grid.arrange(left_side, right_side, ncol = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




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

	Model 1
(Intercept)	19.51^{***}
	(1.10)
polint	1.48^{***}
	(0.34)
R^2	0.11
Adj. R^2	0.10
Num. obs.	165
RMSE	3.42

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

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.

```
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")
```

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

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

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
```

```
update(jags.reg, 200)

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

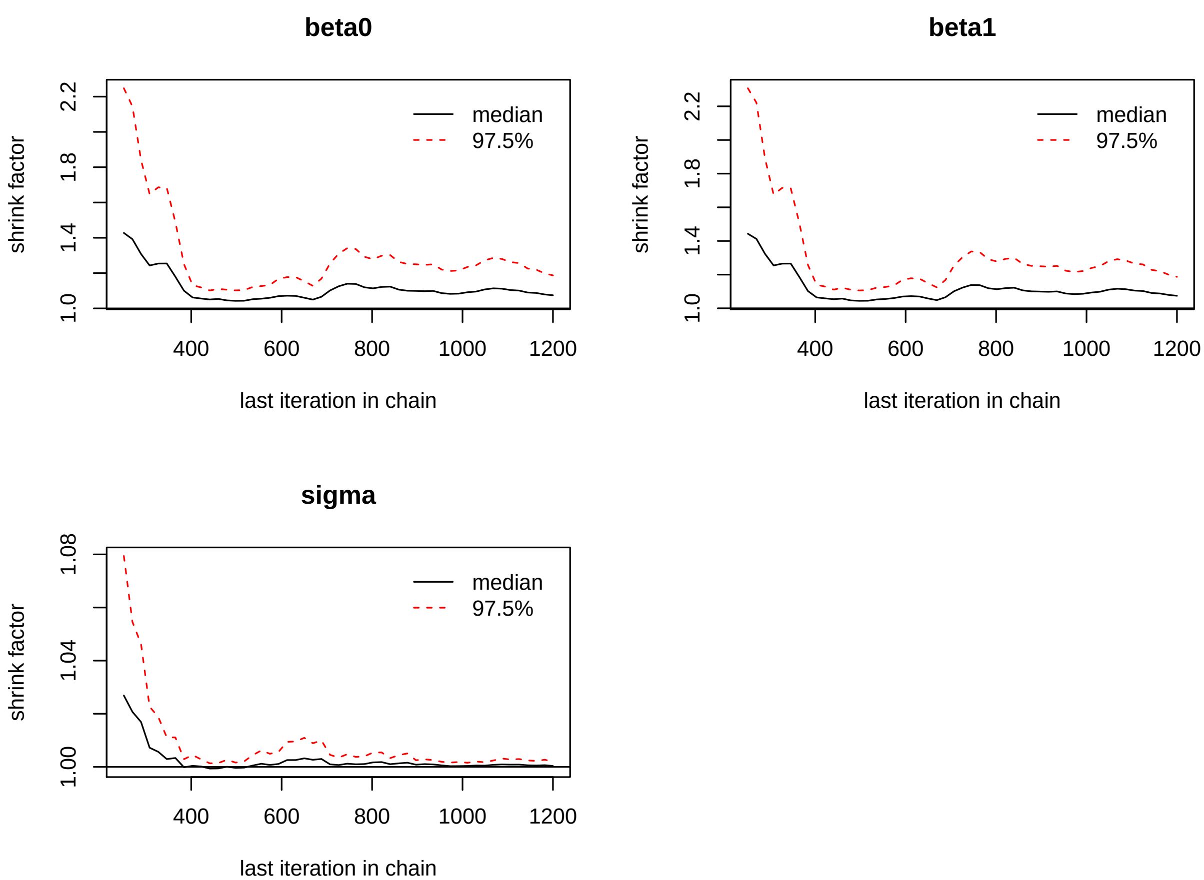
```
  thin = 1
)

jags.reg.out %>%
  summary() %>%
  .$statistics %>%
  kable
```

	Mean	SD	Naive SE	Time-series SE
beta0	19.507713	1.1469954	0.0162210	0.0981903
beta1	1.477109	0.3495683	0.0049436	0.0291069
sigma	3.433150	0.1916856	0.0027108	0.0029300

3 Check Convergence

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

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

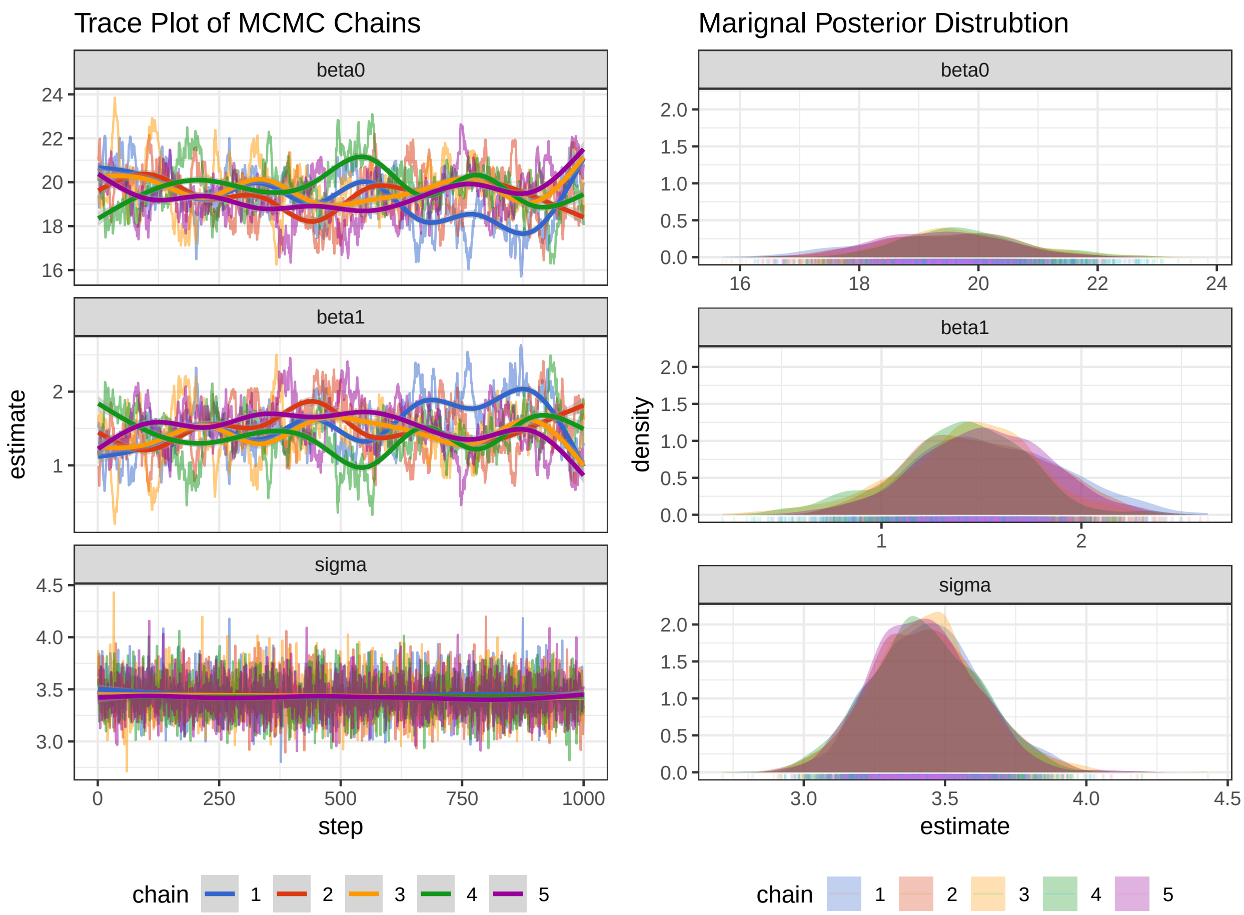


```
#autocorr.plot(jags.reg.out)
```

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[,"beta1"]) %>%
  tibble(p = . > 0) %>%
  tabyl(p) %>%
  kable
```

p	n	percent
TRUE	5000	1

Answer: 100%

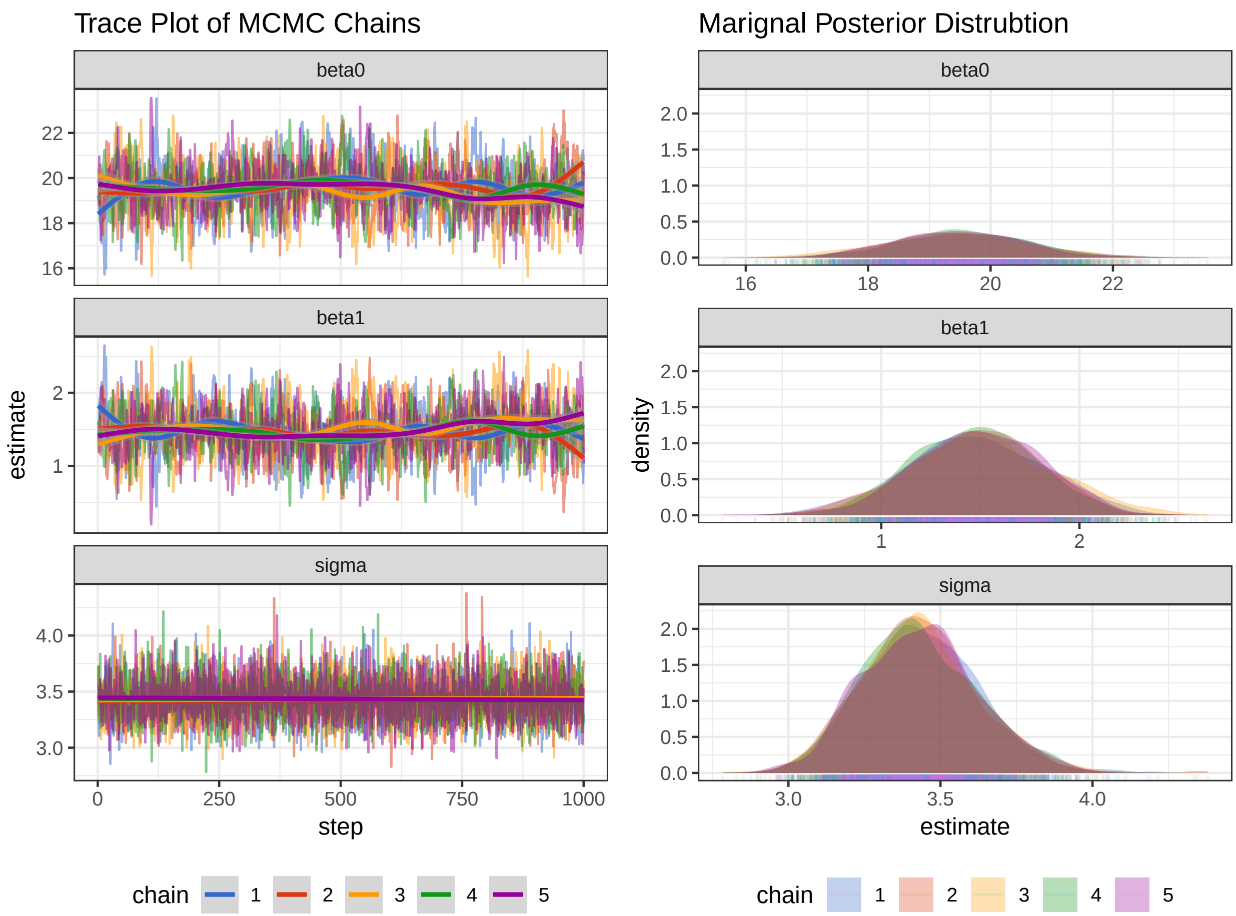
6 Repeat 2-4 with different prior.

```
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(0.01, 0.01)
  sigma <- 1/sqrt(tau)
}"

write(reg.model, "Bayes_Bivariate_Reg_Student_Survey_prior2.bug")

jags.reg2 <- jags.model(
  file = "Bayes_Bivariate_Reg_Student_Survey_prior2.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
```

	Mean	SD	Naive SE	Time-series SE
beta0	19.495866	1.108565	0.0156775	0.0410823
beta1	1.481688	0.339663	0.0048036	0.0125173
sigma	3.432709	0.192654	0.0027245	0.0026634

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



Trace Plot of MCMC Chains
beta0
beta1
sigma
estimate
step
chain 1 2 3 4 5
Marignal Posterior Distrubtion
beta0
beta1
sigma
density
estimate
chain 1 2 3 4 5