2.2 Exercise: priors

Benjamin Rosenbaum

October 25, 2022

Change the prior information for the slope in the linear model (more informative, less informative).

How does the posterior change?

Optional: also change the number of observations (dataset size) as yesterday.

```
rm(list=ls())
library(rstan)
library(coda)

rstan_options(auto_write = TRUE)
options(mc.cores = 4)
```

Models with different priors

We code 4 versions of the linear regression model. Each model has a different prior distribution for the slope `b[2]`.

- model 1: flat prior. nor prior information on `b[2]` given
- model 2: `b[2]~normal(0,10)`
- model 3: `b[2]~normal(0,1)`
- model 4: `b[2]~normal(0,0.1)`

```
stan_code_1 = '
data {
  int n;
  vector[n] x;
  vector[n] y;
}
parameters {
  vector[2] b;
  real<lower=0> sigma;  // standard deviation
}
model {
  // priors
  b[1] ~ normal(0, 10);
  sigma ~ normal(0, 10);
  // likelihood
  y ~ normal(b[1]+b[2]*x, sigma);
}
'

stan_code_2 = '
data {
```

```
  int n;
  vector[n] x;
  vector[n] y;
}
parameters {
  vector[2] b;
  real<lower=0> sigma;  // standard deviation
}
model {
  // priors
  b[1] ~ normal(0, 10);
  b[2] ~ normal(0, 10);
  sigma ~ normal(0, 10);
  // likelihood
  y ~ normal(b[1]+b[2]*x, sigma);
}
'

stan_code_3 = '
data {
  int n;
  vector[n] x;
  vector[n] y;
}
parameters {
  vector[2] b;
  real<lower=0> sigma;  // standard deviation
}
model {
  // priors
  b[1] ~ normal(0, 10);
  b[2] ~ normal(0, 1);
  sigma ~ normal(0, 10);
  // likelihood
  y ~ normal(b[1]+b[2]*x, sigma);
}
'

stan_code_4 = '
data {
  int n;
  vector[n] x;
  vector[n] y;
}
parameters {
  vector[2] b;
  real<lower=0> sigma;  // standard deviation
}
model {
  // priors
  b[1] ~ normal(0, 10);
  b[2] ~ normal(0, 0.1);
  sigma ~ normal(0, 10);
```

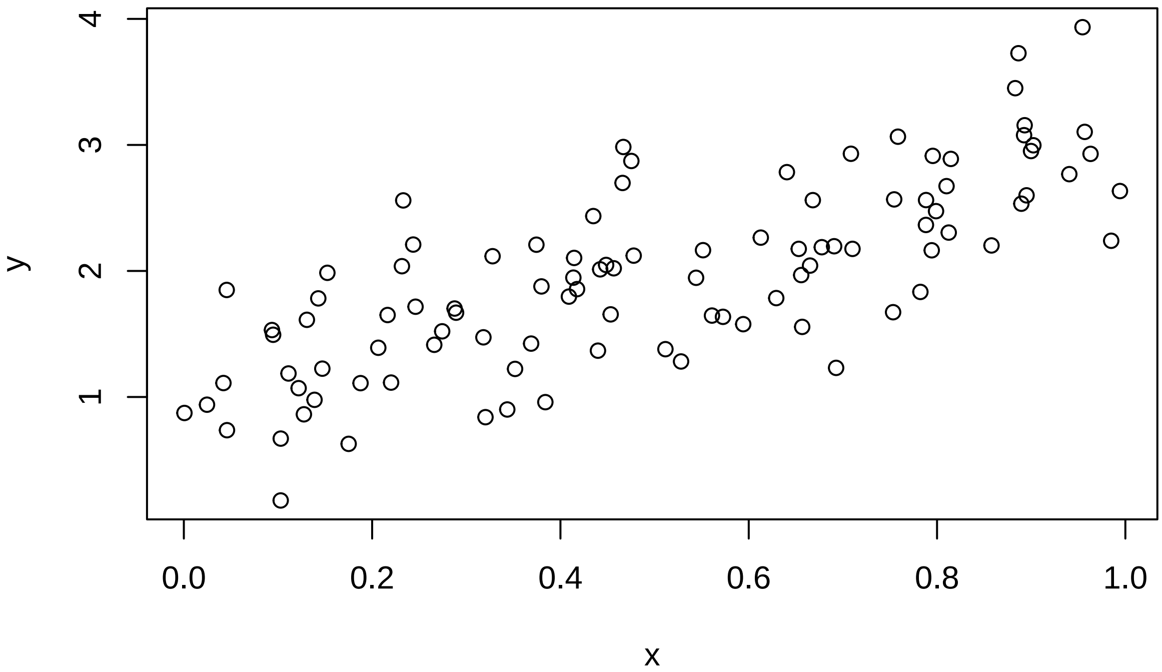
```
  // likelihood
  y ~ normal(b[1]+b[2]*x, sigma);
}
'

stan_model_1 = stan_model(model_code=stan_code_1)
stan_model_2 = stan_model(model_code=stan_code_2)
stan_model_3 = stan_model(model_code=stan_code_3)
stan_model_4 = stan_model(model_code=stan_code_4)
```

Fitting to datasets with varying size

We generate 3 different datasets with varying numbers of observations (10, 100, 1000) and fit all 4 models to each of them.

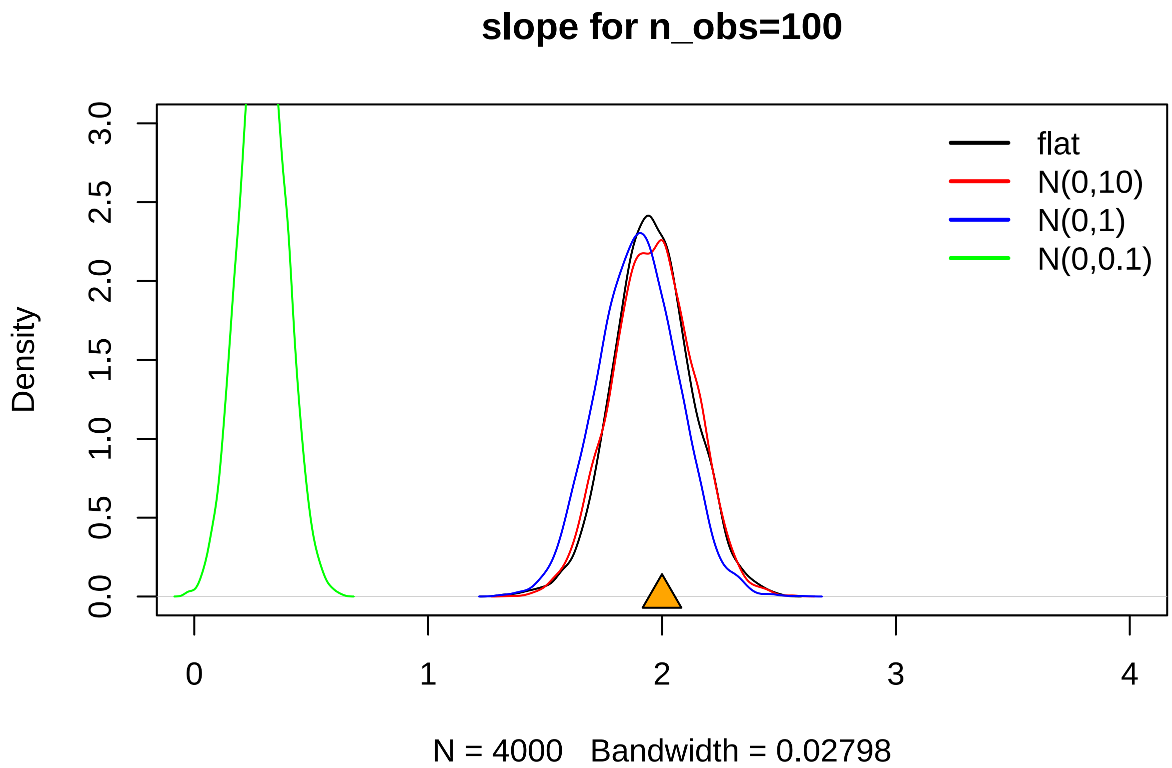
Intermediate dataset

```
set.seed(123) # initiate random number generator for reproducability

n=100

a=1
b=2
sigma=0.5

x = runif(n=n, min=0, max=1)
y = rnorm(n=n, mean=a+b*x, sd=sigma)

df = data.frame(x=x,
                y=y)

plot(df)
```



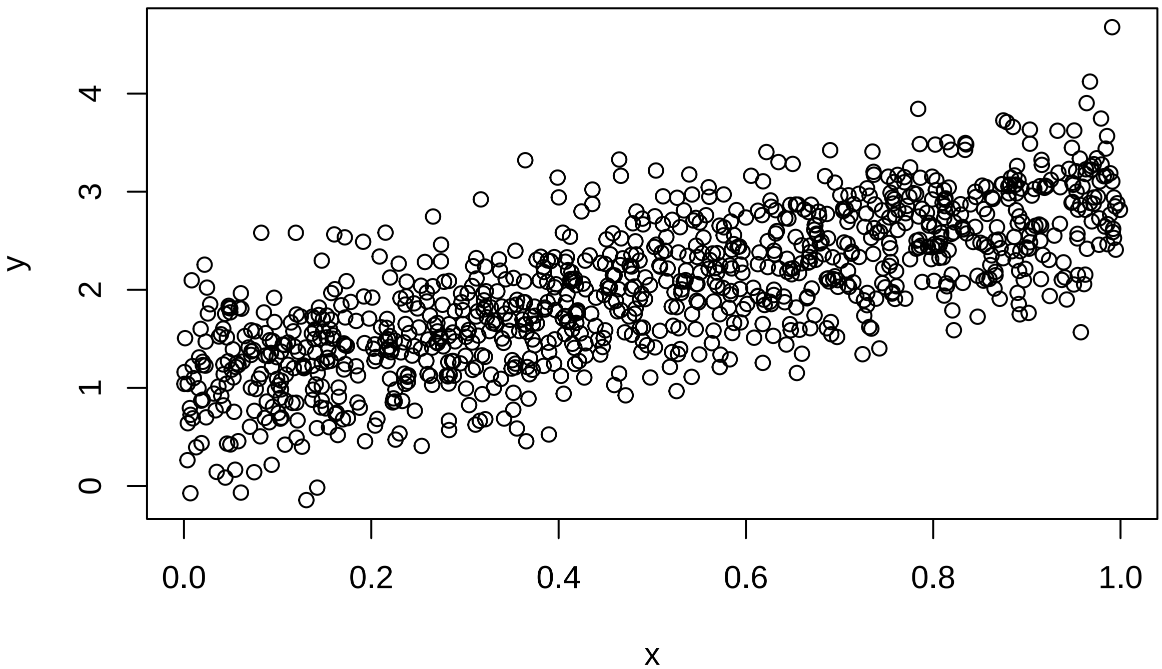

```
data = list(n=n,
            x=df$x,
            y=df$y)

fit_1  = sampling(stan_model_1,
                  data=data)

fit_2  = sampling(stan_model_2,
                  data=data)

fit_3  = sampling(stan_model_3,
                  data=data)

fit_4  = sampling(stan_model_4,
                  data=data)

posterior_1 = as.matrix(fit_1)
posterior_2 = as.matrix(fit_2)
posterior_3 = as.matrix(fit_3)
posterior_4 = as.matrix(fit_4)

density_1=density(posterior_1[, "b[2]"])
density_2=density(posterior_2[, "b[2]"])
density_3=density(posterior_3[, "b[2]"])
density_4=density(posterior_4[, "b[2]"])

par(mfrow=c(1,1))

plot(density_1, xlim=c(0,4), ylim=c(0,3), main="slope for n_obs=100")
lines(density_2, col="red")
lines(density_3, col="blue")
lines(density_4, col="green")
```

```
legend("topright", legend=c("flat","N(0,10)","N(0,1)","N(0,0.1)"), bty="n", lwd=rep(2,4), col=c("black"
points(b,0, pch = 24, cex=2, col="black", bg="orange")
```



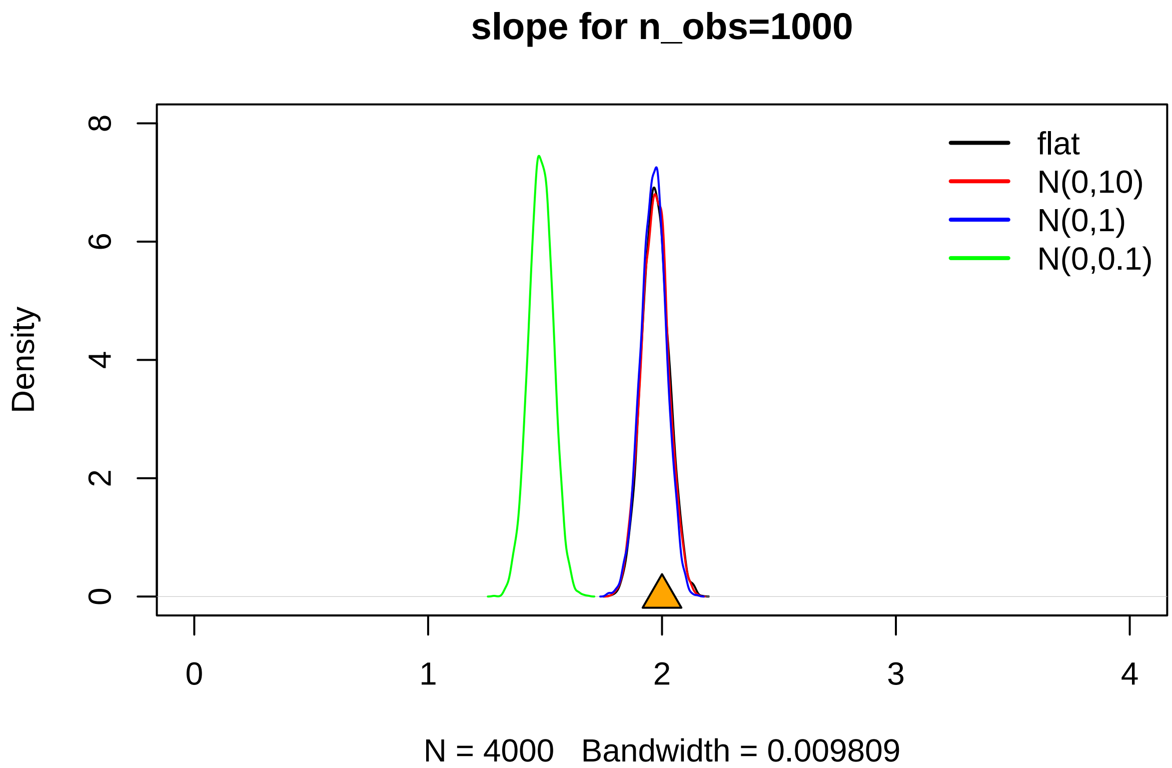
Large dataset

```
set.seed(123) # initiate random number generator for reproducibility

n=1000

a=1
b=2
sigma=0.5

x = runif(n=n, min=0, max=1)
y = rnorm(n=n, mean=a+b*x, sd=sigma)

df = data.frame(x=x,
                y=y)

plot(df)
```



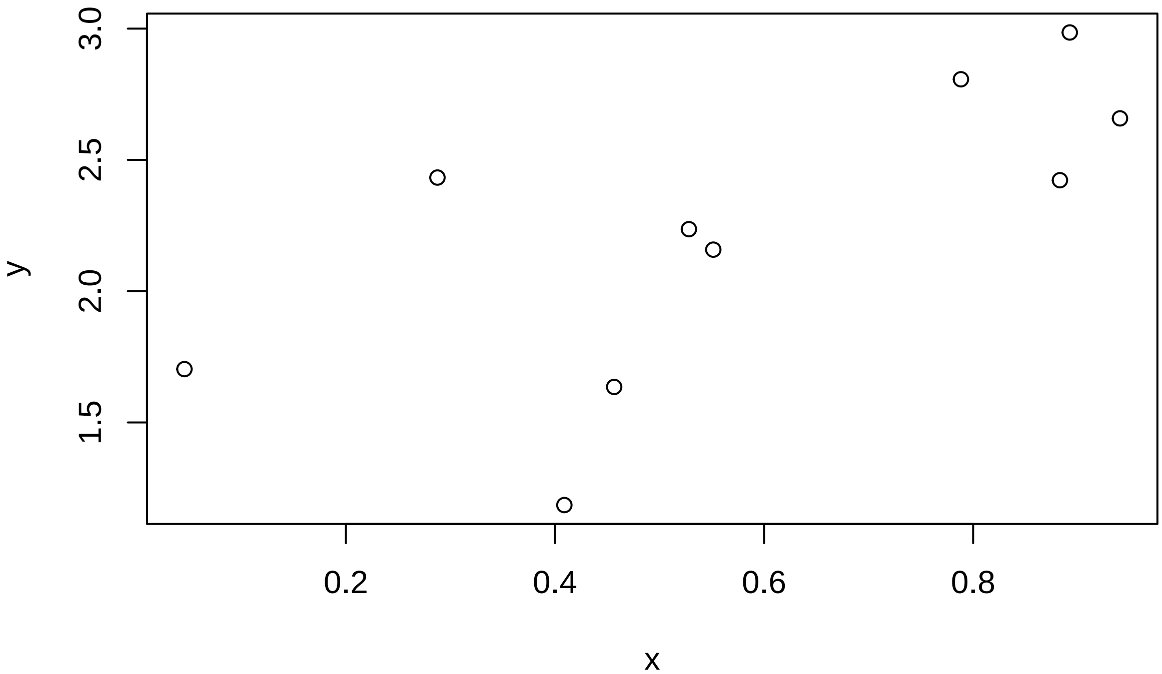

```
data = list(n=n,
            x=df$x,
            y=df$y)

fit_1  = sampling(stan_model_1,
                  data=data)

fit_2  = sampling(stan_model_2,
                  data=data)

fit_3  = sampling(stan_model_3,
                  data=data)

fit_4  = sampling(stan_model_4,
                  data=data)

posterior_1 = as.matrix(fit_1)
posterior_2 = as.matrix(fit_2)
posterior_3 = as.matrix(fit_3)
posterior_4 = as.matrix(fit_4)

density_1=density(posterior_1[, "b[2]"])
density_2=density(posterior_2[, "b[2]"])
density_3=density(posterior_3[, "b[2]"])
density_4=density(posterior_4[, "b[2]"])

par(mfrow=c(1,1))

plot(density_1, xlim=c(0,4), ylim=c(0,8), main="slope for n_obs=1000")
lines(density_2, col="red")
lines(density_3, col="blue")
lines(density_4, col="green")
```

```
legend("topright", legend=c("flat","N(0,10)","N(0,1)","N(0,0.1)"), bty="n", lwd=rep(2,4), col=c("black"
points(b,0, pch = 24, cex=2, col="black", bg="orange")
```



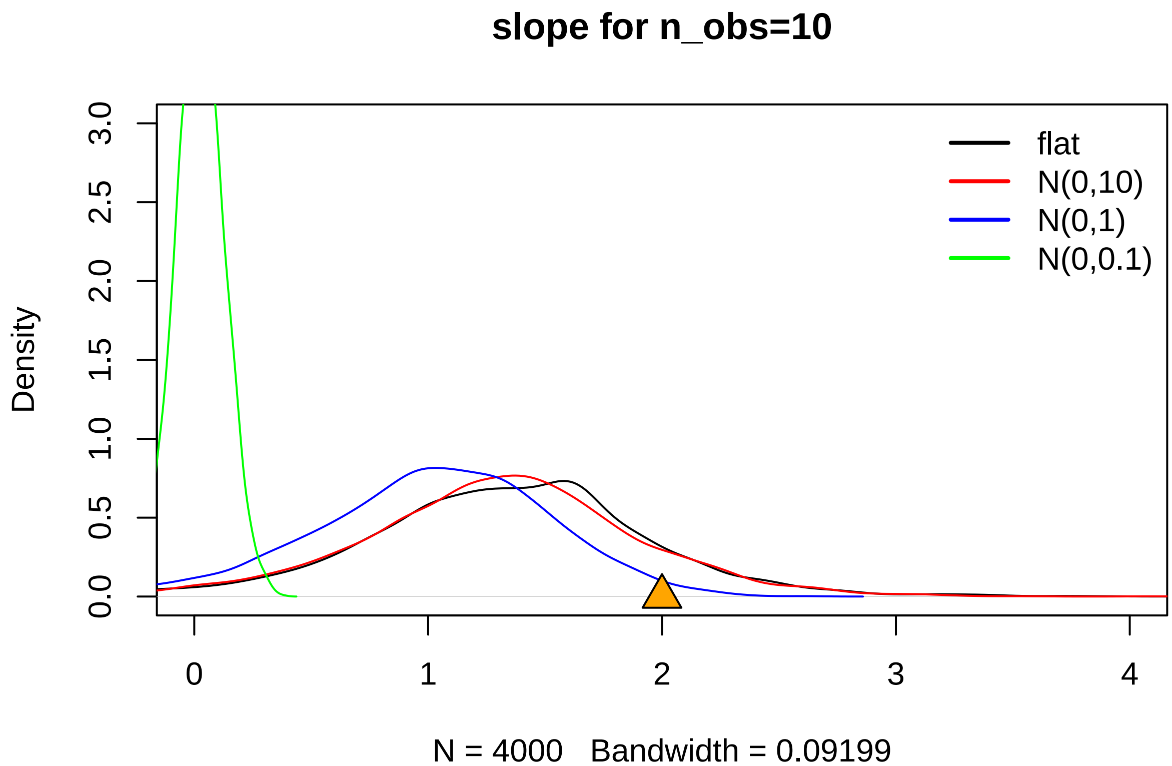
Small dataset

```
set.seed(123) # initiate random number generator for reproducibility

n=10

a=1
b=2
sigma=0.5

x = runif(n=n, min=0, max=1)
y = rnorm(n=n, mean=a+b*x, sd=sigma)

df = data.frame(x=x,
                y=y)

plot(df)
```




```
data = list(n=n,
            x=df$x,
            y=df$y)

fit_1  = sampling(stan_model_1,
                  data=data)

fit_2  = sampling(stan_model_2,
                  data=data)

fit_3  = sampling(stan_model_3,
                  data=data)

fit_4  = sampling(stan_model_4,
                  data=data)

posterior_1 = as.matrix(fit_1)
posterior_2 = as.matrix(fit_2)
posterior_3 = as.matrix(fit_3)
posterior_4 = as.matrix(fit_4)

density_1=density(posterior_1[, "b[2]"])
density_2=density(posterior_2[, "b[2]"])
density_3=density(posterior_3[, "b[2]"])
density_4=density(posterior_4[, "b[2]"])

par(mfrow=c(1,1))

plot(density_1, xlim=c(0,4), ylim=c(0,3), main="slope for n_obs=10")
lines(density_2, col="red")
lines(density_3, col="blue")
lines(density_4, col="green")
```

```
legend("topright", legend=c("flat","N(0,10)","N(0,1)","N(0,0.1)"), bty="n", lwd=rep(2,4), col=c("black"
points(b,0, pch = 24, cex=2, col="black", bg="orange")
```



Conclusions

For the large dataset (`n_obs=1000`), the prior has almost no effect on the posterior distribution.
Only the very informative prior (`normal(0,0.1)`) pulls the posterior estimate towards zero.

For the small dataset (`n_obs=10`), the prior has a strong effect on the posterior distribution.
The more informative the prior, the more the posterior estimate is pulled towards the prior mean of zero.