3_STAN

Stan code

Next we encode similar models in Stan. We start with the presidents, whom we model by student's t likelihood. This is a robust version of normal likelihood, because the heavier tails of the t distribution account better for outliers. Moreover, we let the data determine the heaviness of the tails by fitting the shape parameter nu on the data (in classical stats this parameter is called df - degrees of freedom)

model:

$$y_i \sim student(\nu, \mu, \sigma)$$

$$\nu \sim student(2, 1, 50)$$

$$\mu \sim normal(180, 10)$$

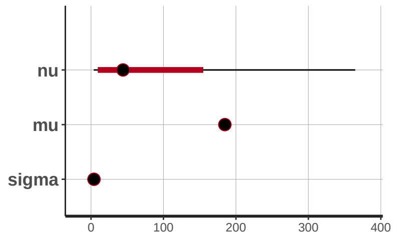
$$\sigma \sim student(4, 0, 5)$$

ülesanne - kirjutage see mudel ümber nii, et see kasutaks normaaljaotuse tõepära ja normaaljaotuse prioreid, kusjuures tehke sigma prior vähem informatiivseks.

```
modelString <- "

data {
  int<lower=0> N; //the nr of data points
  vector[N] heights; //a vector of N datapoints called heights.
}
parameters {
  real<lower=1> nu; //shape parameter cannot be <1
  real mu; //mean
  real<lower=0> sigma; //standard deviation
}
model {
  nu ~ student_t(2, 1, 50); //prior for shape parameter
  mu ~ normal(180, 10); //prior for the mean
  sigma ~ student_t(4, 0, 5); //prior for the SD
  heights ~ student_t(nu, mu, sigma); //regression model
}

"
stanDso <- stan_model(model_code = modelString)

heights <- tibble(value = c(183, 192, 182, 183, 177, 185, 188, 188, 182, 185, 188))

dataList <- list(heights = heights$value, N = length(heights$value))
```

```
fit1 <- sampling(object = stanDso, data = dataList, chains = 3, cores = 3, iter = 1000,
    warmup = 200, thin = 1)

write_rds(fit1, "models/fit1.rds")
```

```
fit1 <- read_rds("models/fit1.rds")
plot(fit1)
```



```
fit1 %>% tidy()
```

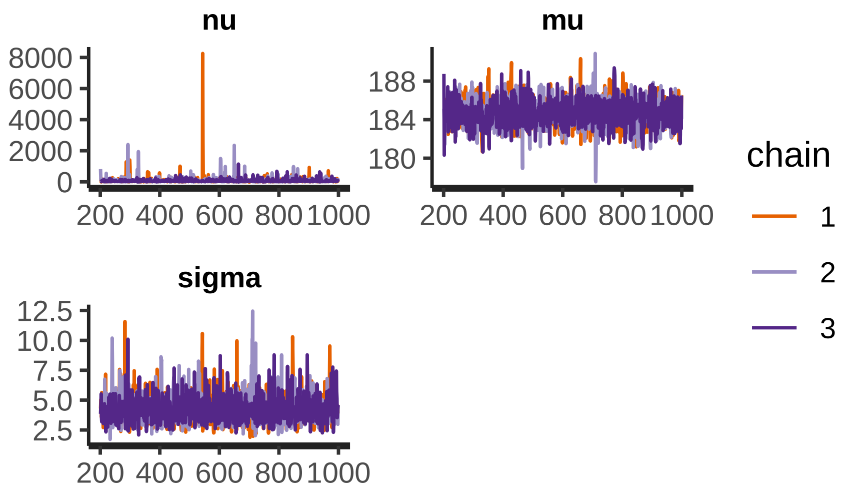
```
## # A tibble: 3 x 3
##   term  estimate std.error
##   <chr>    <dbl>     <dbl>
## 1 nu       83.2     223.
## 2 mu      185.        1.36
## 3 sigma     4.26      1.15
```

```
fit1
```

```
## Inference for Stan model: df64bdb04a46456ae5e57c45f44b989e.
## 3 chains, each with iter=1000; warmup=200; thin=1;
## post-warmup draws per chain=800, total post-warmup draws=2400.
##
##         mean se_mean     sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## nu     83.22    5.51 223.16   3.75  20.77  44.23  85.62 366.07  1640    1
## mu    184.71    0.04   1.36 181.97 183.87 184.72 185.51 187.42  1383    1
## sigma   4.26    0.03   1.15   2.58   3.51   4.06   4.82   6.94  1873    1
## lp__  -21.83    0.05   1.47 -25.72 -22.50 -21.44 -20.80 -20.21   921    1
##
## Samples were drawn using NUTS(diag_e) at Tue Aug  4 17:58:15 2020.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

markov chains

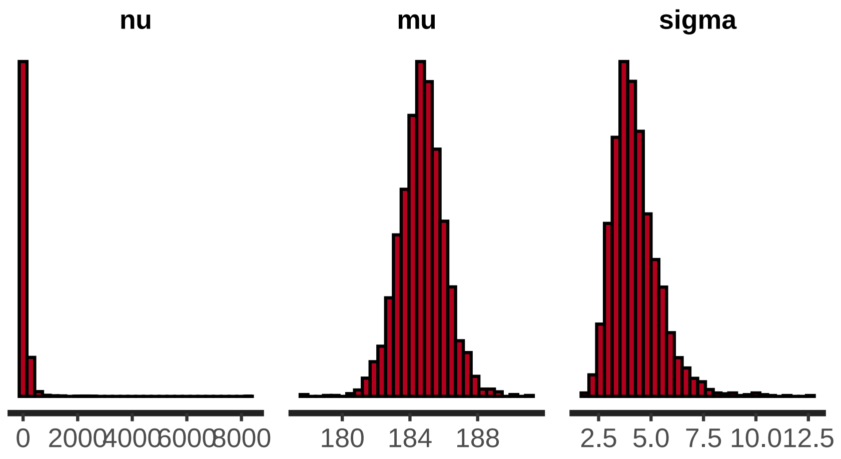
```
traceplot(fit1, nrow = 2)
```



posteriors:

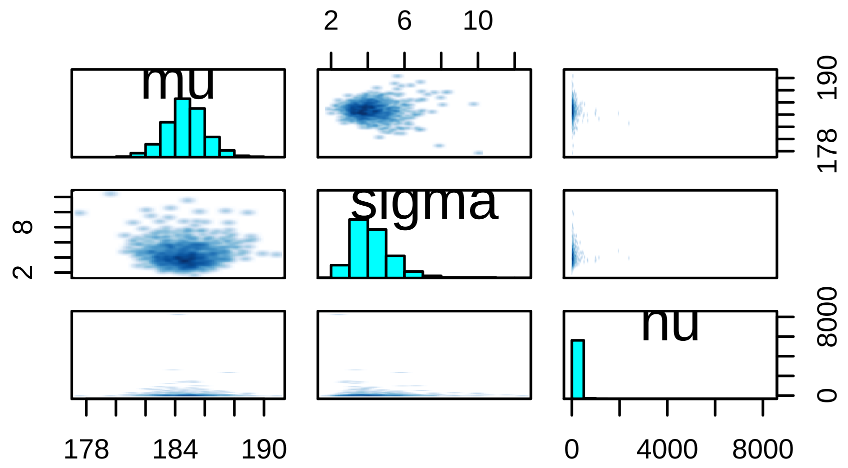
```
# plot(fit1, plotfun = 'hist')
stan_hist(fit1)
```

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

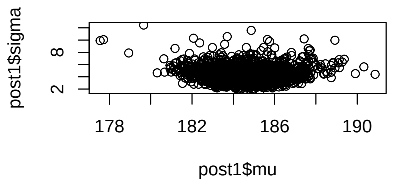


```
pairs(fit1, pars = c("mu", "sigma", "nu"))
```

```
## Warning in KernSmooth::bkde2D(x, bandwidth = bandwidth, gridsize = nbin, :
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'

## Warning in KernSmooth::bkde2D(x, bandwidth = bandwidth, gridsize = nbin, :
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'
```



posteerior samples



```
post1 <- as.data.frame(fit1)
plot(post1$mu, post1$sigma)
```

```
pres_brms <- brm(value ~ 1, data = heights, family = student())
write_rds(pres_brms, "models/pres_brms")
```

```
pres_brms <- read_rds("models/pres_brms")
tidy(pres_brms)
```

```
##          term   estimate std.error      lower      upper
## 1 b_Intercept 184.811496  1.447901 182.461040 187.140324
## 2       sigma   4.350663  1.214018   2.777434   6.537025
## 3          nu  20.378918 13.723315   4.539076  47.855902
## 4        lp__ -37.590361  1.377186 -40.288812 -36.106543
```

excersise – run the same for normal likelihood

expand this into regression

We simulate data:

```
set.seed(12)
data <- tibble(value = c(rnorm(n = 8, mean = 100, sd = 10), rnorm(8, 150, 15)), indeks = rep(c(0,
    1), each = 8))
```

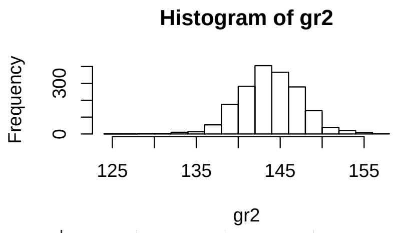
```
t.test(data = data, value ~ indeks)
```

```
##
##  Welch Two Sample t-test
##
## data:  value by indeks
## t = -10.449, df = 13.276, p-value = 8.89e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -60.21584 -39.61759
## sample estimates:
## mean in group 0 mean in group 1
##        93.75789       143.67460
```

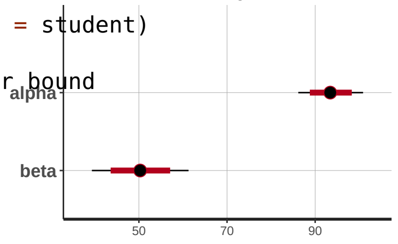
```
lm(data = data, value ~ indeks)
```

```
##
## Call:
## lm(formula = value ~ indeks, data = data)
##
## Coefficients:
## (Intercept)       indeks
##       93.76        49.92
```

```
modelString <- "

data {
  int<lower=0> N;
  vector[N] y;
  vector[N] x;
}
parameters {
  real<lower=1> nu;
  real<lower=0> alpha;
  real beta;
  real<lower=0> sigma;
}
model {
  nu ~ student_t(4, 2, 30); //prior for shape parameter
  alpha ~ normal(100, 50); //prior for the mean
  beta ~ normal(50, 50);
  sigma ~ student_t(4, 0, 20); //prior for the SD
\ty ~ student_t(nu, alpha + beta * x, sigma);
}

"
stanDso <- stan_model(model_code = modelString)

dataList <- list(y = data$value, x = data$indeks, N = nrow(data))

fit2 <- sampling(object = stanDso, data = dataList, chains = 3, cores = 3, iter = 1000,
    warmup = 400, thin = 1)

write_rds(fit2, "models/fit2.rds")

fit2 <- read_rds("models/fit2.rds")
fit2

## Inference for Stan model: da9c5cf4c8c4dcd7b2ad28f7c999ddec.
## 3 chains, each with iter=1000; warmup=400; thin=1;
## post-warmup draws per chain=600, total post-warmup draws=1800.
##
##          mean se_mean    sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## nu      34.12    0.99 33.42   3.06  12.91  25.55  44.09 116.01  1146    1
## alpha   93.46    0.13  3.79  86.17  90.92  93.43  95.91 100.87   831    1
## beta    50.33    0.18  5.37  39.33  47.07  50.29  53.72  61.26   865    1
## sigma    9.72    0.06  2.11   6.31   8.20   9.49  10.89  14.49  1126    1
## lp__   -41.35    0.06  1.62 -45.57 -42.12 -41.01 -40.15 -39.36   718    1
##
## Samples were drawn using NUTS(diag_e) at Tue Aug  4 18:50:47 2020.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).

post2 <- as.data.frame(fit2)
gr2 <- post2$alpha + post2$beta
hist(gr2)
```

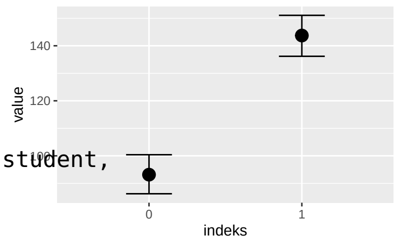


```
plot(fit2, pars = c("alpha", "beta"))
```



```
get_prior(data = data, bf(value ~ indeks, sigma ~ indeks), family = student)

##                   prior     class   coef group resp  dpar nlpar bound
## 1                               b
## 2                               b indeks
## 3 student_t(3, 123, 39) Intercept
## 4          gamma(2, 0.1)        nu
## 5                               b                   sigma
## 6                               b indeks            sigma
## 7   student_t(3, 0, 10) Intercept                   sigma

data1 <- data
data1$indeks <- as.character(data1$indeks)

fit3 <- brm(data = data1, value ~ indeks, family = student, prior = c(prior(student_t(4,
    2, 30), class = nu), prior(normal(100, 50), class = Intercept), prior(normal(50,
    50), class = b), prior(student_t(4, 0, 20), class = sigma)), cores = 3, chains = 3)
write_rds(fit3, "models/fit3.rds")
```

```
fit3 <- read_rds("models/fit3.rds")
conditional_effects(fit3)
```



```
fit4 <- brm(data=data1, bf(value~indeks, sigma~indeks), family = student,

            cores=3, chains = 3)
write_rds(fit4, "models/fit4.rds")
```

```
(fit4 <- read_rds("models/fit4.rds"))
```

```
##  Family: student
##   Links: mu = identity; sigma = log; nu = identity
## Formula: value ~ indeks
##          sigma ~ indeks
##    Data: data1 (Number of observations: 16)
## Samples: 3 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 3000
##
## Population-Level Effects:
##                  Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           93.20      4.20    84.91   101.66 1.00     2071     1376
## sigma_Intercept      2.30      0.34     1.63     3.02 1.00     2065     1317
## indeks1             50.47      5.67    39.04    61.89 1.00     1998     1595
## sigma_indeks1       -0.15      0.45    -1.02     0.79 1.00     2129     2043
##
## Family Specific Parameters:
##    Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## nu    19.51     13.70     2.91    55.80 1.00     2094     1206
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
exp(2.31)
```

```
## [1] 10.07442
```

```
exp(-0.17 + 2.31)
```

```
## [1] 8.499438
```

```
data1 %>% group_by(indeks) %>% summarise(SD = sd(value))
```

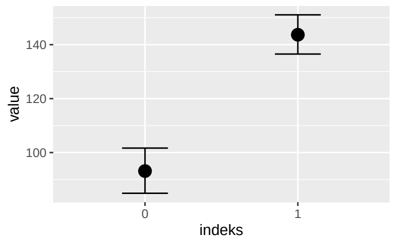
```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```

```
## # A tibble: 2 x 2
##   indeks    SD
##   <chr>  <dbl>
## 1 0      10.6
## 2 1       8.36
```

```
t.test(data = data1, value ~ indeks)
```

```
##
##  Welch Two Sample t-test
##
## data:  value by indeks
## t = -10.449, df = 13.276, p-value = 8.89e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -60.21584 -39.61759
## sample estimates:
## mean in group 0 mean in group 1
##        93.75789       143.67460
```

```
conditional_effects(fit4)
```



model the proportion via binomial distribution

Here we have a covid-19 antibody study of 3330 americans, of whom 50 tested positive. Remember, p is the single parameter in this model.

raw ratio:

```
50/3330
```

```
## [1] 0.01501502
```

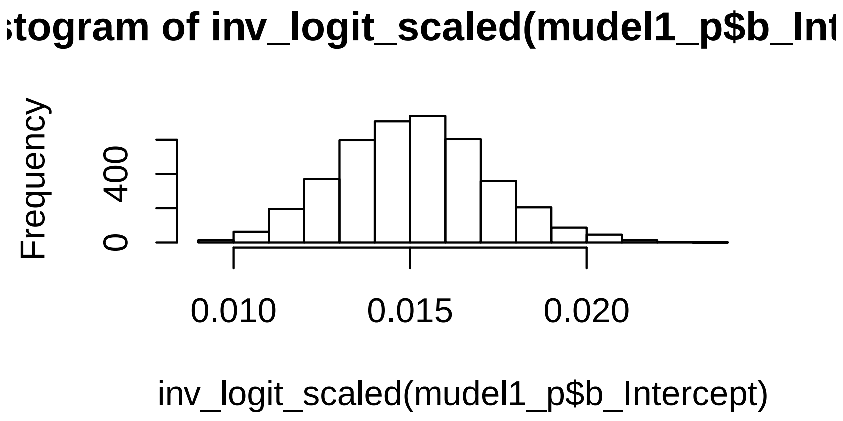
or 1.5%

```
modelString <- "

data {
  int<lower = 0> y;
  int<lower = 0> n;
}
parameters {
  real<lower=0, upper = 1> p;
}
model {
  p ~ beta(1,1);
  y ~ binomial(n, p);
}
```

```
"
stanDso2 <- stan_model(model_code = modelString)

y <- 50
n <- 3330

dataList <- list(y = y, n = n)

stanFit3 <- sampling(object = stanDso2, data = dataList, chains = 3, cores = 3, iter = 1000,
    warmup = 200, thin = 1)

write_rds(stanFit3, "models/stanFit3.rds")
```

```
## # A tibble: 1 x 3
##   term  estimate std.error
##   <chr>    <dbl>     <dbl>
## 1 p       0.0122   0.00264
```

```
mudel1 <- brm(success | trials(3330) ~ 1, data = list(success = 50), family = binomial)
write_rds(mudel1, "models/mudel1.rds")

mudel1 <- read_rds("models/mudel1.rds")
mudel1_p <- posterior_samples(mudel1)
hist(inv_logit_scaled(mudel1_p$b_Intercept))
```



võtame arvesse testi sensitiivsuse ja spetsiifilisuse

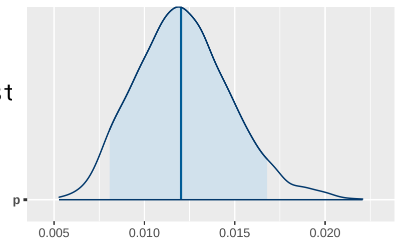
Seda määrati uuringu käigus eraldi, kasutades inimesi, kes konkreetseid proove võtsid ja analüüsisid

```
50/3330 * 103/122 + (1 - 50/3330) * (1 - 399/401)
```

```
## [1] 0.01758925
```

ehk 1.8%
siin jätan priori defineerimata - Stan teeb selle tasaseks prioriks.

```
modelString <- "data {
  int<lower = 0> y_sample;
  int<lower = 0> n_sample;
  real<lower = 0, upper = 1> spec;
  real<lower = 0, upper = 1> sens;
}
parameters {
  real<lower=0, upper = 1> p;
}
model {
  real p_sample = p * sens + (1 - p) * (1 - spec);
  y_sample ~ binomial(n_sample, p_sample);
}"

stanDso3 <- stan_model(model_code = modelString)

y_sample <- 50
n_sample <- 3330
spec <- 399/401
sens <- 103/122

dataList <- list(y_sample = y_sample, n_sample = n_sample, spec = spec, sens = sens)

stanFit3 <- sampling(object = stanDso3, data = dataList, chains = 3, cores = 3, iter = 1000,
    warmup = 200, thin = 1)
write_rds(stanFit3, "models/stanFit3.rds")

stanFit3 <- read_rds("models/stanFit3.rds")
tidy(stanFit3)

## # A tibble: 1 x 3
##   term  estimate std.error
##   <chr>    <dbl>     <dbl>
## 1 p       0.0122   0.00264

mcmc_areas(stanFit3, pars = c("p"), prob = 0.9)

## Warning: ‘expand_scale()‘ is deprecated; use ‘expansion()‘ inst
```



```
modelString <- "data {
  int<lower = 0> y_sample;
  int<lower = 0> n_sample;
  int<lower = 0> y_spec;
  int<lower = 0> n_spec;
  int<lower = 0> y_sens;
  int<lower = 0> n_sens;
}
parameters {
  real<lower=0, upper = 1> p;
  real<lower=0, upper = 1> spec;
  real<lower=0, upper = 1> sens;
}
model {
  real p_sample = p * sens + (1 - p) * (1 - spec);
  y_sample ~ binomial(n_sample, p_sample);
  y_spec ~ binomial(n_spec, spec);
  y_sens ~ binomial(n_sens, sens);
}"

stanDso4 <- stan_model(model_code = modelString)

y_sample <- 50
n_sample <- 3330
y_spec <- 399
n_spec <- 401
y_sens <- 103
n_sens <- 122

dataList <- list(y_sample = y_sample, n_sample = n_sample, y_spec = y_spec, n_spec = n_spec,
    y_sens = y_sens, n_sens = n_sens)

stanFit4 <- sampling(object = stanDso4, data = dataList, chains = 3, cores = 3, iter = 1000,
    warmup = 200, thin = 1)

write_rds(stanFit4, "models/stanFit4.rds")
```

```
stanFit4 <- read_rds("models/stanFit4.rds")
tidy(stanFit4)
```

```
## # A tibble: 3 x 3
##   term  estimate std.error
##   <chr>    <dbl>     <dbl>
## 1 p       0.0103   0.00472
## 2 spec    0.993    0.00350
## 3 sens    0.838    0.0328
```

lets slap some priors on p, sens and spec. Especially for p I'm putting a strong prior that says that most likely incidence is 1% and incidences over 3% are very unlikely. I am vaguer for sensitivity and specificity where the prior has heavy tails allowing for a wide range of possibilities.

```
modelString <- "data {
  int<lower = 0> y_sample;
  int<lower = 0> n_sample;
  int<lower = 0> y_spec;
  int<lower = 0> n_spec;
  int<lower = 0> y_sens;
  int<lower = 0> n_sens;
}
parameters {
  real<lower=0, upper = 1> p;
  real<lower=0, upper = 1> spec;
  real<lower=0, upper = 1> sens;
}
model {
  real p_sample = p * sens + (1 - p) * (1 - spec);
  p ~ normal(0.01, 0.01);
  spec ~ student_t(6, 0.9, 0.1);
  sens ~ student_t(3, 0.8, 0.1);
  y_sample ~ binomial(n_sample, p_sample);
  y_spec ~ binomial(n_spec, spec);
  y_sens ~ binomial(n_sens, sens);
}"

stanDso5 <- stan_model(model_code = modelString)

y_sample <- 50
n_sample <- 3330
y_spec <- 399
n_spec <- 401
y_sens <- 103
n_sens <- 122

dataList <- list(y_sample = y_sample, n_sample = n_sample, y_spec = y_spec, n_spec = n_spec,
    y_sens = y_sens, n_sens = n_sens)

stanFit5 <- sampling(object = stanDso5, data = dataList, chains = 3, cores = 3, iter = 1000,
    warmup = 300, thin = 1)
write_rds(stanFit5, "models/stanFit5.rds")

stanFit5 <- read_rds("models/stanFit5.rds")
tidy(stanFit5)
```

```
## # A tibble: 3 x 3
##   term  estimate std.error
##   <chr>    <dbl>     <dbl>
## 1 p       0.0106   0.00420
## 2 spec    0.993    0.00321
## 3 sens    0.833    0.0338
```

```
0.01059813 - 1.96 * 0.004201813
```

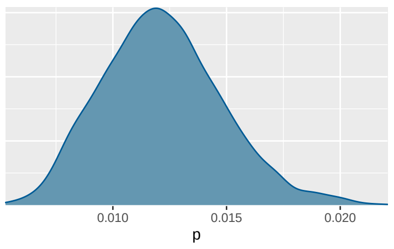
```
## [1] 0.002362577
```

lower bound for the p is about 0.2% incidence
higher bound is about 1.9%

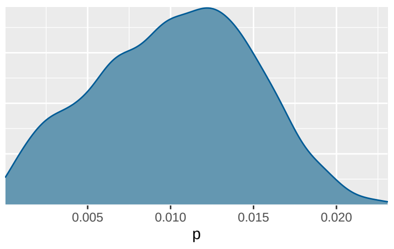
```
0.01059813 + 1.96 * 0.004201813
```

```
## [1] 0.01883368
```

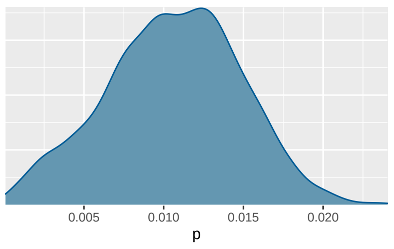
```
mcmc_dens(stanFit3, pars = c("p"))
```



```
mcmc_dens(stanFit4, pars = c("p"))
```



```
mcmc_dens(stanFit5, pars = c("p"))
```


0.005
0.010
0.015
0.020
p