STA 602 LAB 4

William Tirone

```r
require(rstan)
```

```
Loading required package: rstan
```

```
Loading required package: StanHeaders
```

```
Loading required package: ggplot2
```

```
rstan (Version 2.21.7, GitRev: 2e1f913d3ca3)
```

```
For execution on a local, multicore CPU with excess RAM we recommend calling
options(mc.cores = parallel::detectCores()).
To avoid recompilation of unchanged Stan programs, we recommend calling
rstan_options(auto_write = TRUE)
```

```r
require(tidyverse)
```

```
Loading required package: tidyverse
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v tibble  3.1.8      v dplyr   1.0.10
v tidyr   1.2.1      v stringr 1.4.1
v readr   2.1.3      v forcats 0.5.2
v purrr   0.3.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x tidyr::extract() masks rstan::extract()
x dplyr::filter()  masks stats::filter()
x dplyr::lag()     masks stats::lag()
```

```r
require(rstanarm)
```

```
Loading required package: rstanarm
Loading required package: Rcpp
This is rstanarm version 2.21.3
- See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
- Default priors may change, so it's safest to specify priors, even if equivalent to the defa
- For execution on a local, multicore CPU with excess RAM we recommend calling
  options(mc.cores = parallel::detectCores())

Attaching package: 'rstanarm'

The following object is masked from 'package:rstan':

    loo
```

```r
require(magrittr)
```

```
Loading required package: magrittr

Attaching package: 'magrittr'

The following object is masked from 'package:purrr':

    set_names

The following object is masked from 'package:tidyr':

    extract

The following object is masked from 'package:rstan':

    extract
```

prior selection

```
create_df <- function(post_draws, prior_draws){
  post_draws <- data.frame(post_draws)
  post_draws$distribution <- "posterior"

  prior_draws <- data.frame(prior_draws)
  colnames(prior_draws) <- "alpha"
  prior_draws$distribution <- "prior"

  dat <- rbind(post_draws, prior_draws)
  return(dat)
}
set.seed(689934)

alpha <- 1
beta <- -0.25
sigma <- 1

N <- 5
x <- array(runif(N, 0, 2), dim=N)
y <- array(rnorm(N, beta * x + alpha, sigma), dim=N)
```

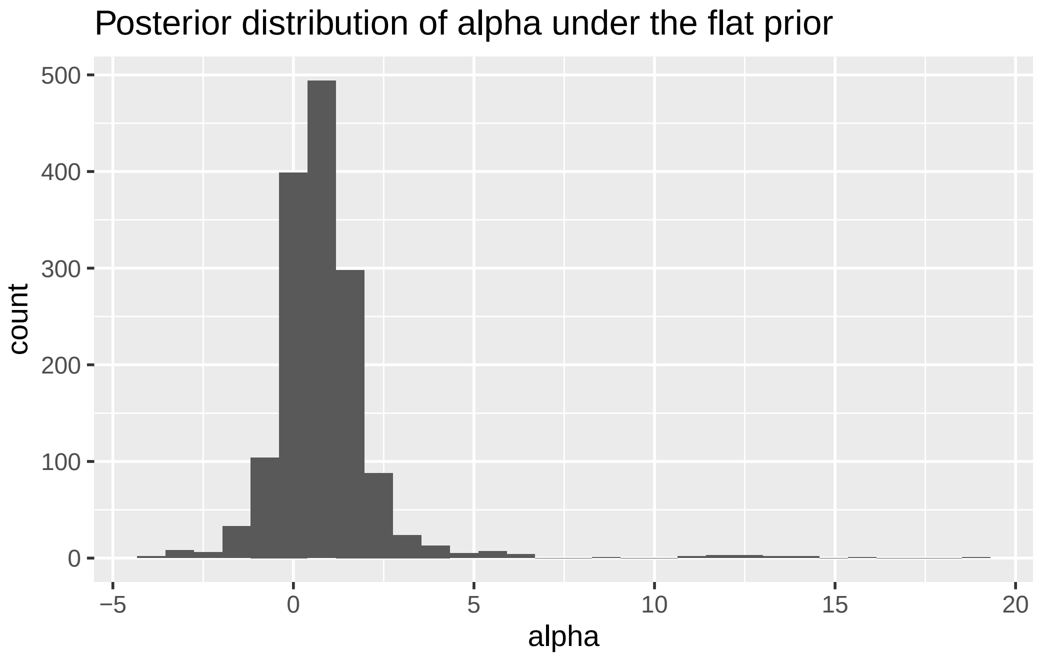
Flat priors:

```
stan_dat <- list(y = y, x=x, N=N)
fit.flat <- stan(file = "lab-04-flat_prior.stan", data = stan_dat,
                 chains = 1, refresh = 0, iter = 2000, warmup = 500, seed=48)
```

```
Trying to compile a simple C file

Running /usr/lib64/R/bin/R CMD SHLIB foo.c
gcc -m64 -I"/usr/include/R" -DNDEBUG   -I"/usr/lib64/R/library/Rcpp/include/"  -I"/usr/lib64/
In file included from /usr/lib64/R/library/RcppEigen/include/Eigen/Dense:1,
                 from /usr/lib64/R/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.l
                 from <command-line>:
/usr/lib64/R/library/RcppEigen/include/Eigen/Core:82:12: fatal error: new: No such file or di
   82 |   #include <new>
      |            ^~~~~
```

```
compilation terminated.
make: *** [/usr/lib64/R/etc/Makeconf:168: foo.o] Error 1

Warning: There were 77 divergent transitions after warmup. See
https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
to find out why this is a problem and how to eliminate them.

Warning: Examine the pairs() plot to diagnose sampling problems

Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians
Running the chains for more iterations may help. See
https://mc-stan.org/misc/warnings.html#bulk-ess

Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and ta
Running the chains for more iterations may help. See
https://mc-stan.org/misc/warnings.html#tail-ess
```

```
## Trying to compile a simple C file
alpha.flat <- as.matrix(fit.flat, pars = "alpha")
beta.flat <- as.matrix(fit.flat, pars = "beta")
ggplot(alpha.flat %>% as.data.frame, aes(x = alpha)) +
  geom_histogram(bins = 30) +
  labs(title = "Posterior distribution of alpha under the flat prior")
```



```
print(fit.flat, pars = c("alpha","beta"))
```

```
Inference for Stan model: lab-04-flat_prior.
1 chains, each with iter=2000; warmup=500; thin=1;
post-warmup draws per chain=1500, total post-warmup draws=1500.

      mean se_mean   sd  2.5%   25%  50%  75% 97.5% n_eff Rhat
alpha 0.87    0.13 1.66 -1.36  0.17 0.71 1.33  3.77   164    1
beta  0.19    0.10 1.22 -2.18 -0.09 0.34 0.71  1.65   158    1

Samples were drawn using NUTS(diag_e) at Mon Oct  3 14:36:37 2022.
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).
```

Exercise 1:

Write down the posterior means of a and b. Give 95% credible intervals for each. Comment on how diffuse the posteriors of a and b are. Considering the amount of data and the type of prior we choose, do the results seem surprising? Explain.

both a and b are quite diffuse given the lack of data and the estimate viewed from stan above. Proof of a and b are quite difficult and will be completed later.

From class:

$$E(\alpha|Y) = (n\sum x_i^2 - (\sum x_i)^2)^{-1}(\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i) Var(\alpha|Y) = (n - \frac{(\sum x_i)^2}{\sum x_i})^{-1}$$

another flat prior UNIF(-10,10)

```
stan_dat <- list(y = y, x=x, N=N, lb = -10, ub = 10)
fit.unif <- stan(file = "lab-04-unif_prior.stan", data = stan_dat,
                 chains = 1, refresh = 0, iter = 2000, warmup = 500, seed=48)
```

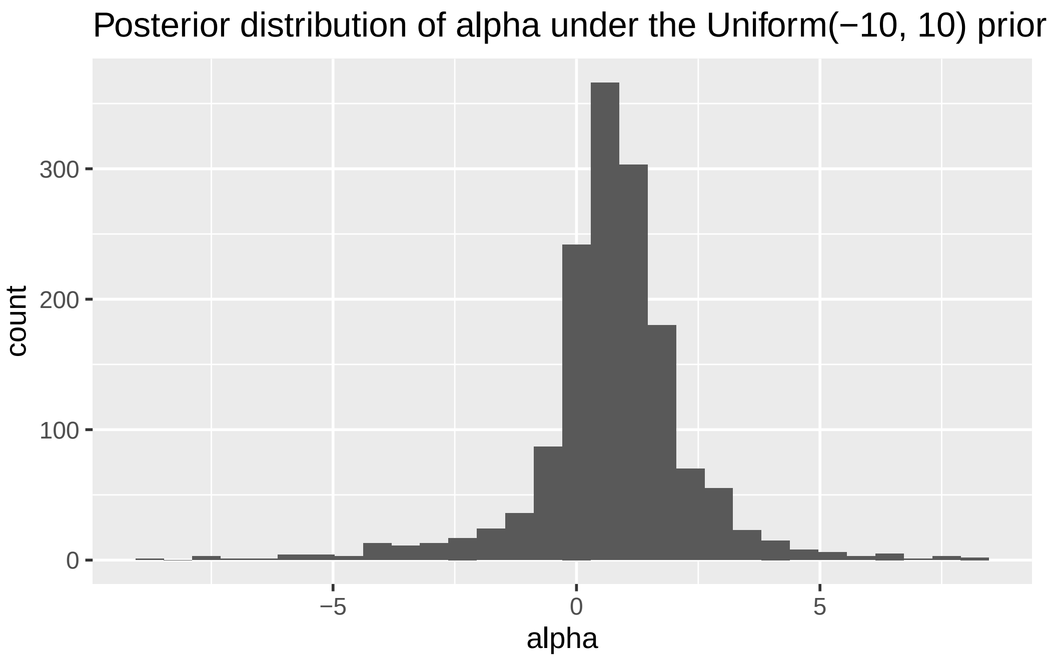
```
Trying to compile a simple C file
```

```
Running /usr/lib64/R/bin/R CMD SHLIB foo.c
gcc -m64 -I"/usr/include/R" -DNDEBUG   -I"/usr/lib64/R/library/Rcpp/include/"  -I"/usr/lib64/
In file included from /usr/lib64/R/library/RcppEigen/include/Eigen/Dense:1,
                 from /usr/lib64/R/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.
                 from <command-line>:
/usr/lib64/R/library/RcppEigen/include/Eigen/Core:82:12: fatal error: new: No such file or di
   82 |   #include <new>
      |            ^~~~~
compilation terminated.
make: *** [/usr/lib64/R/etc/Makeconf:168: foo.o] Error 1
```

```
Warning: There were 3 divergent transitions after warmup. See
https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
to find out why this is a problem and how to eliminate them.
```

```
Warning: Examine the pairs() plot to diagnose sampling problems
```

```r
alpha.unif <- as.matrix(fit.unif, pars = c("alpha"))
beta.unif <- as.matrix(fit.unif, pars = c("beta"))
ggplot(alpha.unif %>% as.data.frame, aes(x = alpha)) +
  geom_histogram(bins = 30) +
  labs(title = "Posterior distribution of alpha under the Uniform(-10, 10) prior")
```



posterior of the UNIF(-10,10) prior

```
print(fit.unif, pars = c("alpha"))
```

```
Inference for Stan model: lab-04-unif_prior.
1 chains, each with iter=2000; warmup=500; thin=1;
post-warmup draws per chain=1500, total post-warmup draws=1500.

      mean se_mean   sd  2.5%  25%  50%  75% 97.5% n_eff Rhat
alpha 0.73    0.11 1.63 -3.49 0.13 0.76 1.45  4.01   209    1

Samples were drawn using NUTS(diag_e) at Mon Oct  3 14:37:24 2022.
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).
```

Consider N(0,1) prior

```
stan_dat <- list(y = y, x=x, N=N)
fit.norm <- stan(file = "lab-04-normal_prior.stan", data = stan_dat,
                 chains = 1, refresh = 0, iter = 2000, warmup = 500, seed=49)
```

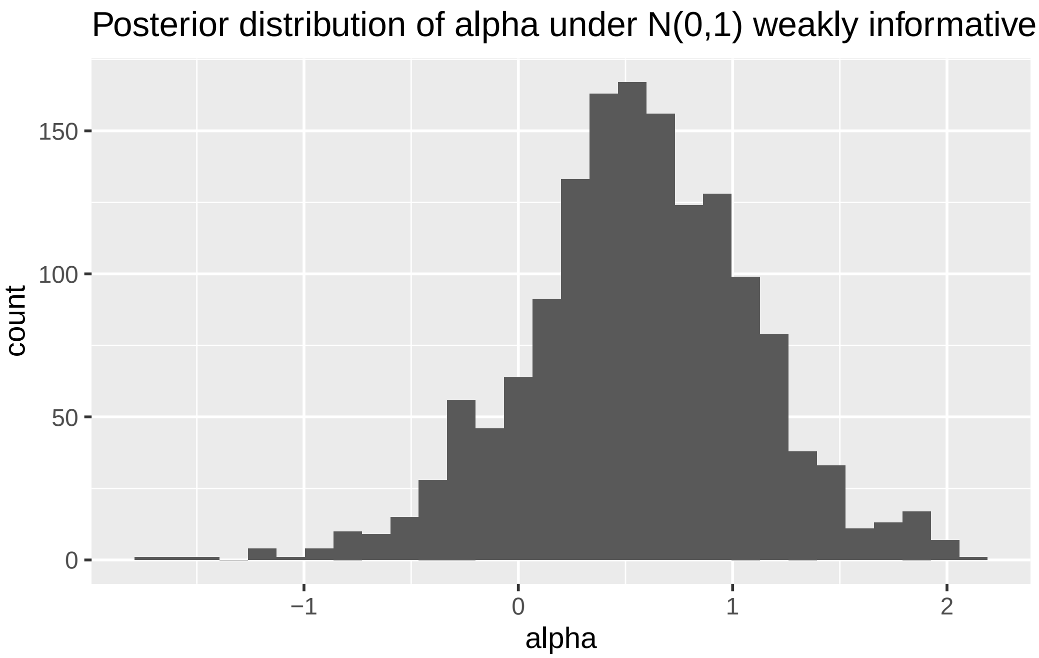
```
Trying to compile a simple C file
```

```
Running /usr/lib64/R/bin/R CMD SHLIB foo.c
gcc -m64 -I"/usr/include/R" -DNDEBUG   -I"/usr/lib64/R/library/Rcpp/include/"  -I"/usr/lib64/
In file included from /usr/lib64/R/library/RcppEigen/include/Eigen/Dense:1,
                 from /usr/lib64/R/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.
                 from <command-line>:
/usr/lib64/R/library/RcppEigen/include/Eigen/Core:82:12: fatal error: new: No such file or d
   82 |   #include <new>
      |            ^~~~~
compilation terminated.
make: *** [/usr/lib64/R/etc/Makeconf:168: foo.o] Error 1
```

```
Warning: There were 2 divergent transitions after warmup. See
https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
to find out why this is a problem and how to eliminate them.
```

```
Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Trying to compile a simple C file
alpha.norm<- as.matrix(fit.norm, pars = c("alpha"))
ggplot(alpha.norm %>% as.data.frame, aes(x = alpha)) +
  geom_histogram(bins = 30) +
  labs(title = "Posterior distribution of alpha under N(0,1) weakly informative prior")
```



```
print(fit.norm, pars = c("alpha","beta"))
```

```
Inference for Stan model: lab-04-normal_prior.
1 chains, each with iter=2000; warmup=500; thin=1;
post-warmup draws per chain=1500, total post-warmup draws=1500.

      mean se_mean   sd  2.5%  25%  50%  75% 97.5% n_eff Rhat
alpha 0.56    0.02 0.54 -0.52 0.24 0.56 0.91  1.66   581    1
beta  0.39    0.02 0.43 -0.43 0.13 0.38 0.66  1.28   587    1

Samples were drawn using NUTS(diag_e) at Mon Oct  3 14:38:11 2022.
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).
```

Exercise 2:

Again proof is very difficult, but looking at the estimated alpha and beta from stan, this is less diffuse with a lower standard deviation but still not totally informative.

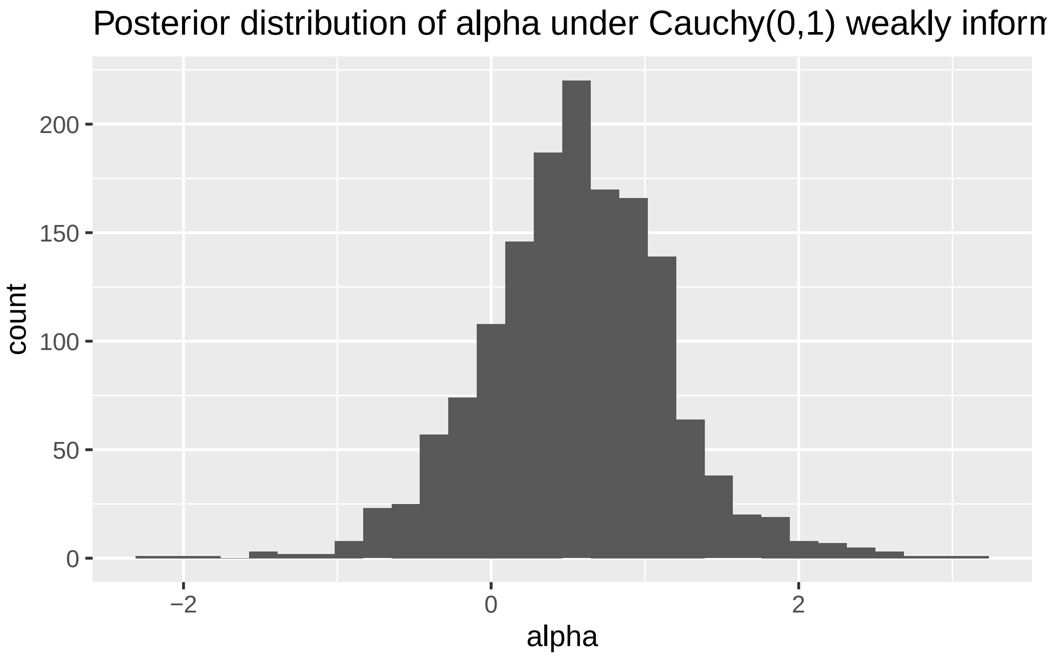
Heavy-tailed prior ~ cauchy(0,1)

```
stan_dat <- list(y = y, x=x, N=N)
fit.cauchy <- stan(file = "lab-04-cauchy_prior.stan",data = stan_dat,
                   chains = 1, refresh = 0, iter = 2000, warmup = 500, seed=55)
```

```
Trying to compile a simple C file
```

```
Running /usr/lib64/R/bin/R CMD SHLIB foo.c
gcc -m64 -I"/usr/include/R" -DNDEBUG   -I"/usr/lib64/R/library/Rcpp/include/"  -I"/usr/lib64/
In file included from /usr/lib64/R/library/RcppEigen/include/Eigen/Dense:1,
                 from /usr/lib64/R/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.h
                 from <command-line>:
/usr/lib64/R/library/RcppEigen/include/Eigen/Core:82:12: fatal error: new: No such file or di
   82 |   #include <new>
      |            ^~~~~
compilation terminated.
make: *** [/usr/lib64/R/etc/Makeconf:168: foo.o] Error 1
```

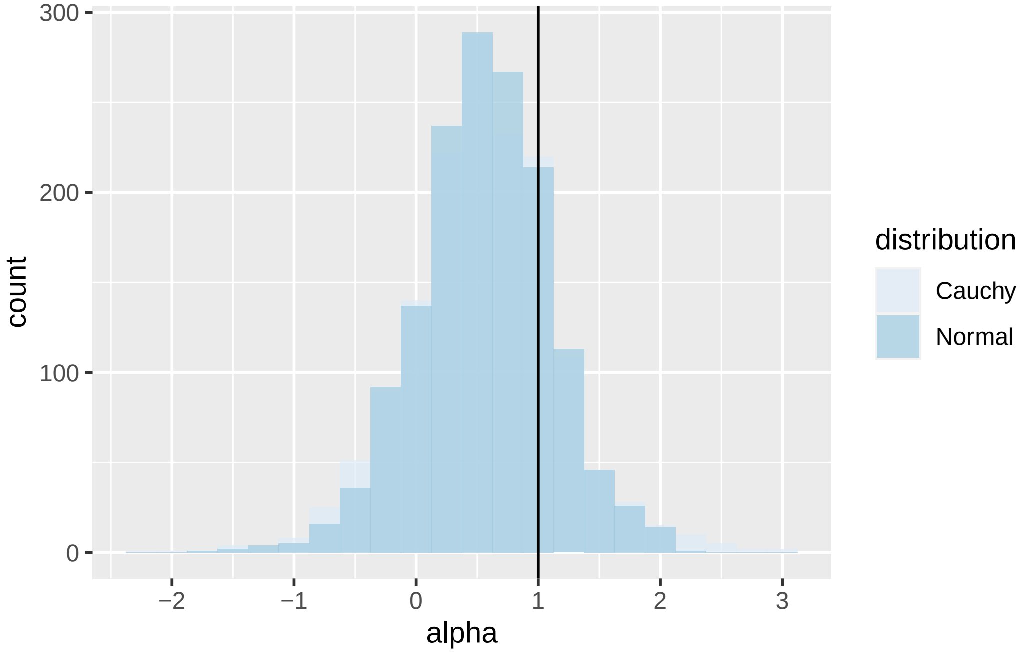
```
## Trying to compile a simple C file
alpha.cauchy<- as.matrix(fit.cauchy, pars = c("alpha"))
ggplot(alpha.cauchy %>% as.data.frame, aes(x = alpha)) +
  geom_histogram(bins = 30) +
  labs(title = "Posterior distribution of alpha under Cauchy(0,1) weakly informative prior
```



```
print(fit.cauchy, pars = c("alpha","beta"))
```

```
Inference for Stan model: lab-04-cauchy_prior.
1 chains, each with iter=2000; warmup=500; thin=1;
post-warmup draws per chain=1500, total post-warmup draws=1500.

      mean se_mean   sd  2.5%  25%  50%  75% 97.5% n_eff Rhat
alpha 0.55    0.03 0.61 -0.67 0.19 0.55 0.94  1.83   488    1
beta  0.39    0.02 0.46 -0.53 0.11 0.38 0.66  1.33   563    1

Samples were drawn using NUTS(diag_e) at Mon Oct  3 14:38:56 2022.
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).
```

```
plot_dat <- create_df(alpha.norm, alpha.cauchy) %>%
  mutate(distribution = if_else(distribution == "posterior", "Normal","Cauchy"))

ggplot(plot_dat, aes(alpha, fill = distribution)) +
  geom_histogram(binwidth = 0.25, alpha = 0.7, position = "identity")+
  geom_vline(xintercept = alpha) +
  scale_fill_brewer()
```



Exercise 3:

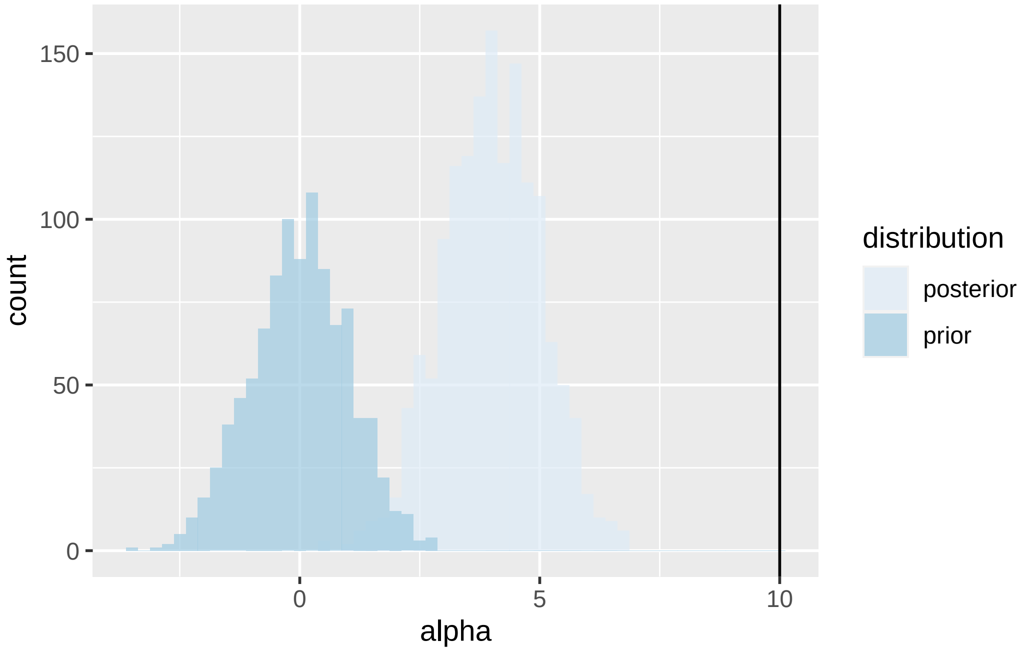
Would you say that a Cauchy prior is more or less informative than a Normal prior (assume that their inter-quartile ranges are comparable)? Compared to the Normal prior, how much probability mass does Cauchy prior put in the tail regions?

The cauchy puts much more mass in the tails so I think this is saying that there's a greater chance we could see extreme values for a and b. This seems less informative than the normal prior because it is less precise.

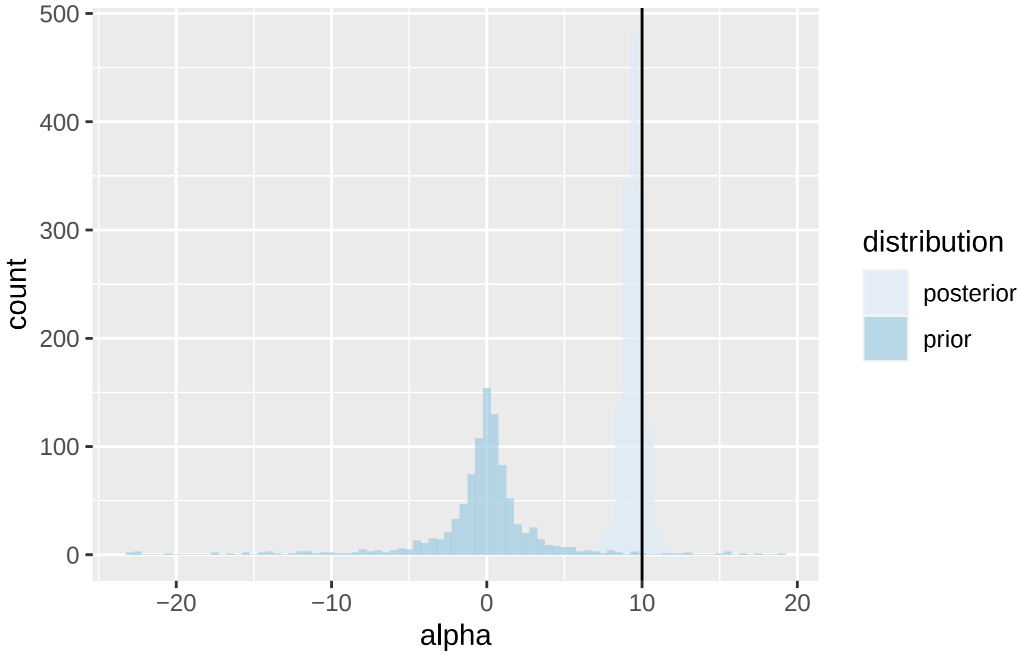
Sensitivity to prior selection:

```
alpha <- 10
N <- 10
x <- runif(N, 0, 2)
y <- rnorm(N, beta * x + alpha, sigma)

stan_dat <- list(y = y, x=x, N=N)
```

```r
fit.norm <- stan(file = "lab-04-normal_prior.stan", data = stan_dat,
                 chains = 1, refresh = 0, iter = 2000, warmup = 500, seed=49)
alpha.norm<- as.matrix(fit.norm, pars = c("alpha"))
prior_draws <- rnorm(1000, 0, 1)
plot_dat <- create_df(alpha.norm, prior_draws)

ggplot(plot_dat, aes(alpha, fill = distribution)) +
  geom_histogram(binwidth = 0.25, alpha = 0.7, position = "identity")+
  geom_vline(xintercept = alpha) +
  scale_fill_brewer()
```



```r
stan_dat <- list(y = y, x=x, N=N)
fit.cauchy <- stan(file = "lab-04-cauchy_prior.stan",data = stan_dat,
                   chains = 1, refresh = 0, iter = 2000, warmup = 500, seed=55)
alpha.cauchy<- as.matrix(fit.cauchy, pars = c("alpha"))
prior_draws <- rcauchy(1000, 0, 1)
prior_draws <- prior_draws[abs(prior_draws) < 25]
plot_dat <- create_df(alpha.cauchy, prior_draws)

ggplot(plot_dat, aes(alpha, fill = distribution)) +
  geom_histogram(binwidth = .5, alpha = 0.7, position = "identity")+
  geom_vline(xintercept = alpha) +
  scale_fill_brewer()
```



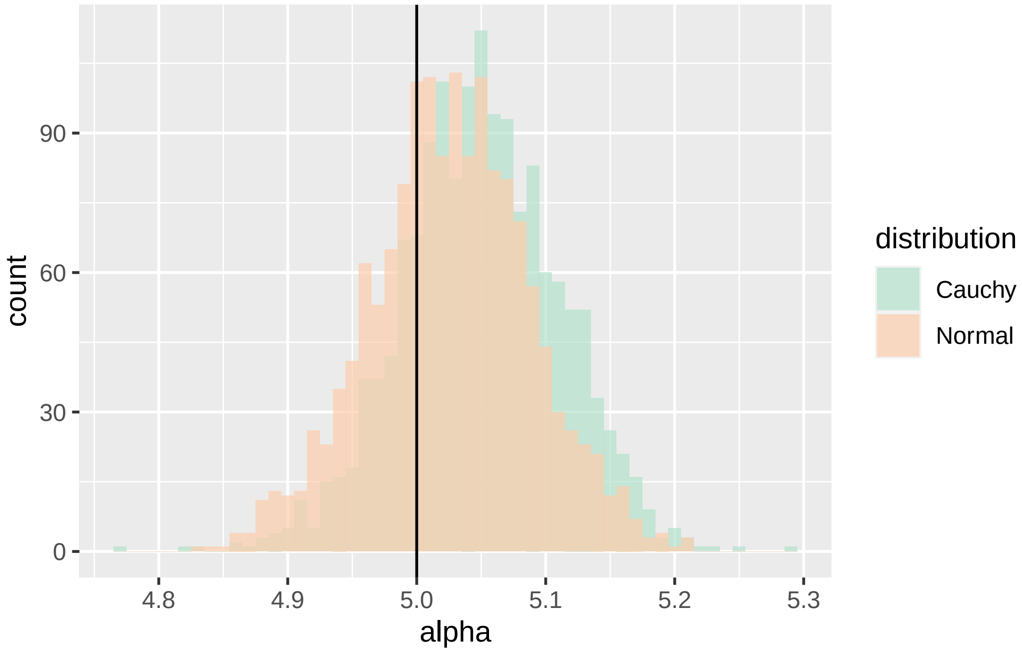
Exercise 4:

```
alpha <- 5
beta <- -0.25
sigma <- 1

N <- 1000
x <- array(runif(N, 0, 2), dim=N)
y <- array(rnorm(N, beta * x + alpha, sigma), dim=N)


stan_dat <- list(y = y, x=x, N=N)
fit.norm <- stan(file = "lab-04-normal_prior.stan", data = stan_dat,
                 chains = 1, refresh = 0, iter = 2000, warmup = 500, seed=49)
alpha.norm<- as.matrix(fit.norm, pars = c("alpha"))

prior_draws <- rnorm(1000, 0, 1)
```

```r
plot_dat <- create_df(alpha.norm, prior_draws)

stan_dat <- list(y = y, x=x, N=N)
fit.cauchy <- stan(file = "lab-04-cauchy_prior.stan",data = stan_dat,
                   chains = 1, refresh = 0, iter = 2000, warmup = 500, seed=55)
alpha.cauchy<- as.matrix(fit.cauchy, pars = c("alpha"))

prior_draws <- rcauchy(1000, 0, 1)
prior_draws <- prior_draws[abs(prior_draws) < 25]
plot_dat <- create_df(alpha.cauchy, prior_draws)

print(fit.cauchy, pars = c("alpha","beta"))
```

```
Inference for Stan model: lab-04-cauchy_prior.
1 chains, each with iter=2000; warmup=500; thin=1;
post-warmup draws per chain=1500, total post-warmup draws=1500.

       mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
alpha  5.05       0 0.06  4.93  5.01  5.05  5.09  5.17   560 1.01
beta  -0.30       0 0.05 -0.40 -0.34 -0.30 -0.26 -0.20   494 1.01

Samples were drawn using NUTS(diag_e) at Mon Oct  3 14:39:03 2022.
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).
```

```r
plot_dat <- create_df(alpha.norm, alpha.cauchy) %>%
  mutate(distribution = if_else(distribution == "posterior", "Normal","Cauchy"))

ggplot(plot_dat, aes(alpha, fill = distribution)) +
  geom_histogram(binwidth = 0.01, alpha = 0.7, position = "identity")+
  geom_vline(xintercept = alpha) +
  scale_fill_brewer(palette='Pastel2')
```



They perform fairly similarly with N=5 and alpha = 1. It looks like the normal is a little more precise. Comparing to the true alpha above, it looks like the cauchy is a little bit closer but still not great, and the normal performs poorly.

I also tried increasing the N to 1000 (plotted above) and while they look similar I think the cauchy is performing better. So it looks like with a small sample size, the posterior is more affected by the prior while having a large sample size will let the sampling model dominate the posterior.

Exercise 5:

If we thought for some reason that we could observe extreme values, we might want the heavier tails of the cauchy to potentially capture that info. If you don't have a lot of prior knowledge, you would choose the cauchy as well. However, if you had expert knowledge maybe the normal prior with a more precise estimate would be better.

However, the large sample sizes will lead to a similar posterior.

Exercise 6:

The scale of your prior would likely be based on either expert knowledge or the results from previous experiments or studies. For example, if you were measuring the weight of animals you would want your mean and variance to be similar to the past observed values to appropriately estimate the population.

```
set.seed(123);
theta <- 0.3;
N <- 10;
y <- rbinom(N, 1, theta)
theta.mle <- sum(y)/N
stan_dat <- list(y = y,N=N)
fit.bayes.prob <- stan(file = "lab-04-prob.stan", data = stan_dat, refresh = 0, iter = 200
```

```
Trying to compile a simple C file
```

```
Running /usr/lib64/R/bin/R CMD SHLIB foo.c
gcc -m64 -I"/usr/include/R" -DNDEBUG   -I"/usr/lib64/R/library/Rcpp/include/"  -I"/usr/lib64/
In file included from /usr/lib64/R/library/RcppEigen/include/Eigen/Dense:1,
                 from /usr/lib64/R/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.h
                 from <command-line>:
/usr/lib64/R/library/RcppEigen/include/Eigen/Core:82:12: fatal error: new: No such file or di
   82 |   #include <new>
      |            ^~~~~
compilation terminated.
make: *** [/usr/lib64/R/etc/Makeconf:168: foo.o] Error 1
```

```
## Trying to compile a simple C file
print(fit.bayes.prob, pars = c("theta", "eta"))
```

```
Inference for Stan model: lab-04-prob.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

       mean se_mean   sd  2.5%   25%   50%  75% 97.5% n_eff Rhat
theta  0.41    0.00 0.14  0.16  0.31  0.41 0.51  0.69  1179 1.01
eta   -0.38    0.02 0.61 -1.64 -0.78 -0.38 0.02  0.80  1222 1.00

Samples were drawn using NUTS(diag_e) at Mon Oct  3 14:39:46 2022.
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).
```

Exercise 7

The posterior beta would be a beta(1,1) prior times the “true” sampling model of beta(3,7) so posterior is beta(4,8)

$$\frac{\alpha - 1}{\alpha + \beta - 2} = \frac{3}{4 + 8 - 2} = .30$$

I am very unsure of how to calculate the mode based on the stan output above even with the lab instructions.

Exercise 8

This is not a prior proper as the real line doesn’t integrate to 1, so it’s not a pdf. The bayesian procedure does still work, and it does result in a proper posterior I believe because of some normalizing constant. I’m not clear on which conditions, I would guess under the conditions that you can find a normalizing constant to find a closed-form posterior.

Exercise 9

```r
fit.logodds <- stan(file = "lab-04-log_odds.stan", data = stan_dat, refresh = 0, iter = 20
```

```
Trying to compile a simple C file
```

```
Running /usr/lib64/R/bin/R CMD SHLIB foo.c
gcc -m64 -I"/usr/include/R" -DNDEBUG   -I"/usr/lib64/R/library/Rcpp/include/"  -I"/usr/lib64
In file included from /usr/lib64/R/library/RcppEigen/include/Eigen/Dense:1,
                 from /usr/lib64/R/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.
                 from <command-line>:
/usr/lib64/R/library/RcppEigen/include/Eigen/Core:82:12: fatal error: new: No such file or di
   82 |   #include <new>
      |            ^~~~~
compilation terminated.
make: *** [/usr/lib64/R/etc/Makeconf:168: foo.o] Error 1
```

```r
## Trying to compile a simple C file
print(fit.logodds, pars = c("theta", "eta"))
```

```
Inference for Stan model: lab-04-log_odds.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

       mean se_mean   sd  2.5%   25%   50% 75% 97.5% n_eff Rhat
theta  0.39    0.00 0.15  0.14  0.29  0.39 0.5  0.69  1407    1
eta   -0.47    0.02 0.68 -1.81 -0.91 -0.47 0.0  0.80  1392    1

Samples were drawn using NUTS(diag_e) at Mon Oct  3 14:40:29 2022.
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).
```

the induced prior is a uniform(0,1/2) and is a proper prior.