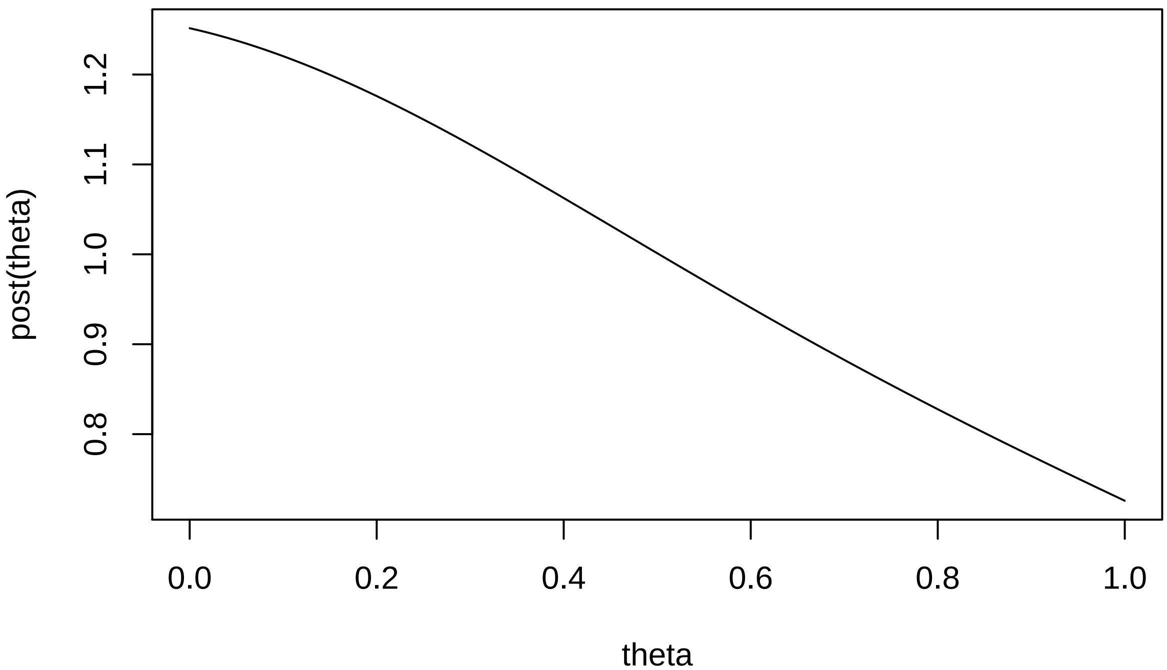
Homework 3

1. A. The posterior is given by $\frac{P(y|\theta)P(\theta)}{\int_\theta P(y|\theta)P(\theta)} = \frac{\prod_y (1+(y-\theta)^2)^{-1}}{\int_0^1 \prod_y (1+(y-\theta)^2)^{-1}}$. I computed the integral numerically to get 0.003391051 so we get the following curve

```
post <- function (th){
     y <- c(-2, -1, 0, 1.5, 2.5)
     p <- prod (cauchy_lik (y, th))/0.003391051
     p}
cauchy_lik <- function (y, th){
  cl <- (1 + (y - th)^2)^(-1)
  cl}
post <- Vectorize(post)
curve(post, 0, 1, xname = "theta")
```



B. $\log P(\theta|y) = \text{constant} - \sum_y \log(1 + (y-\theta)^2)$

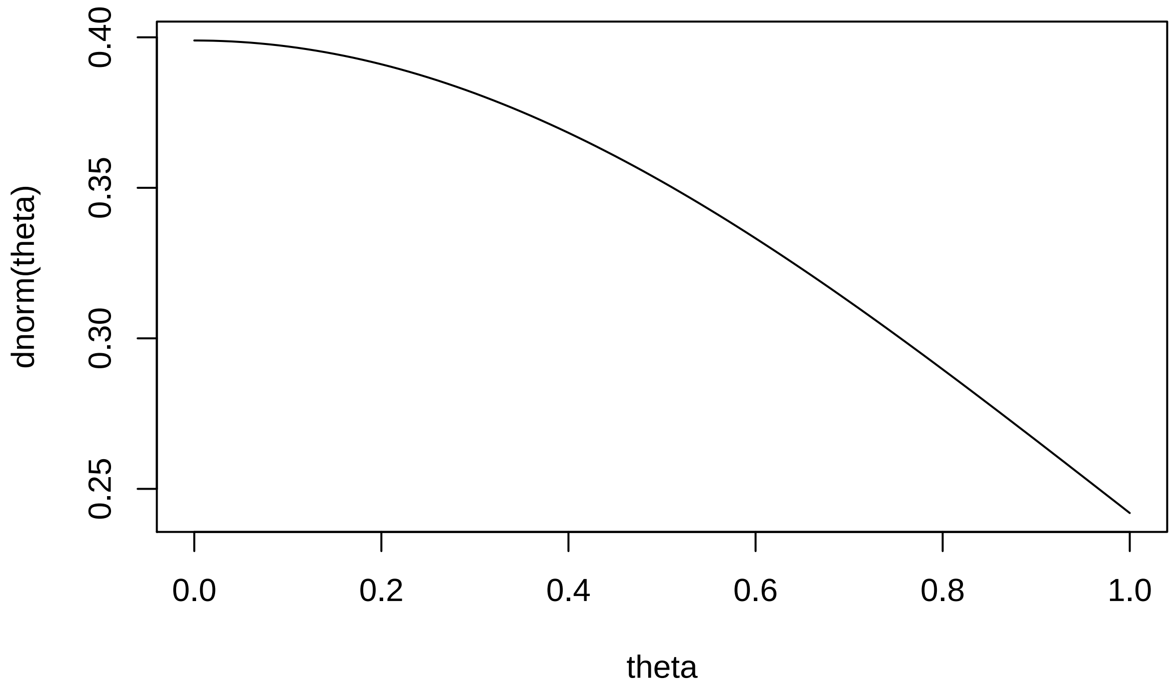
So the first derivative is $\sum_y \frac{2(y-\theta)}{1+(y-\theta)^2}$

And the second derivative is (using the product rule) $\sum_y \frac{2(y-\theta)^2-2}{(1+(y-\theta)^2)^2)}$

C. We use Newton's method and start with theta = 0. First derivative at 0 is -.18 and the second is -1.32 giving us a posterior mode of -.136

D. The second derivative at the mode is -1.37 so the observed information is 1.37. Therefore the Normal approximation is (-.18, 0.73)

```
curve(dnorm, mean=0.18, sd=0.73, 0, 1, xname = "theta")
```



This seems slightly less heavy-tailed than the exact posterior

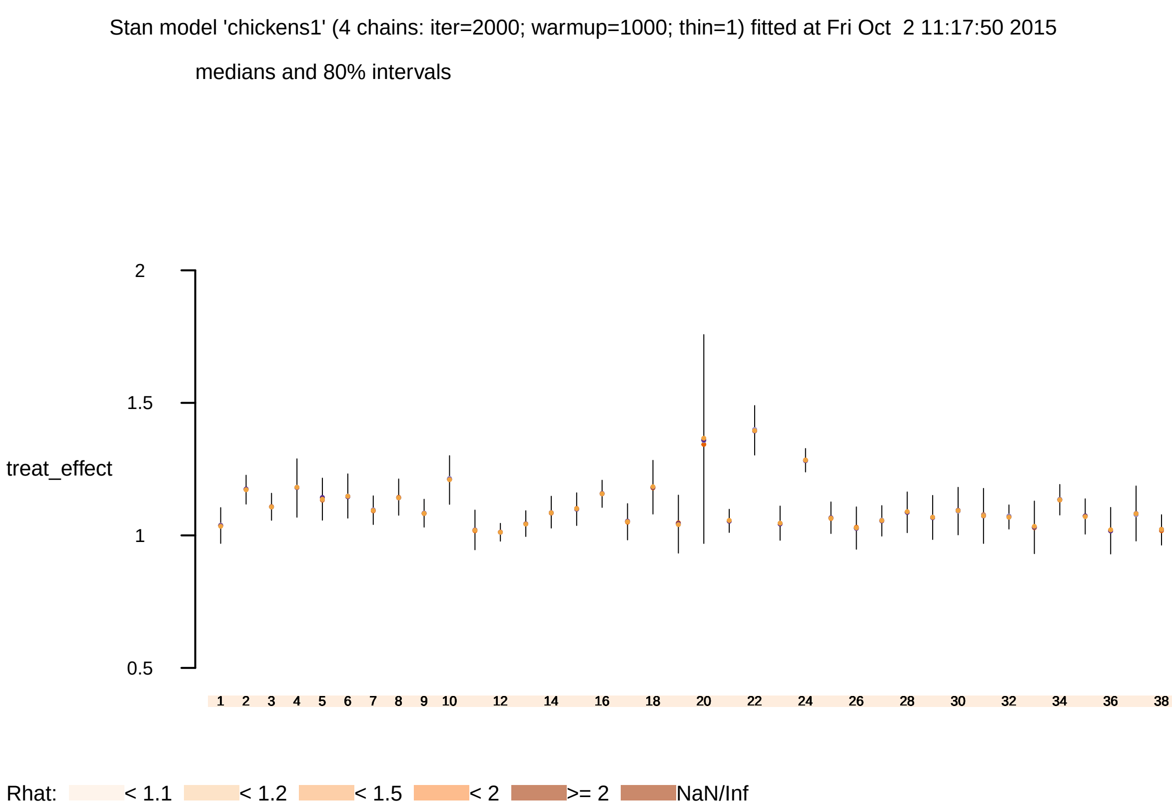
2. A. We fit several models in stan. We assumed that the mean of each difference $\text{mean}_j \sim \mathcal{N}(\text{treat}_j, \sigma_j$ where $\sigma_j = (N_j * \text{SE}_j)^2$ We first fit a model where there was a noninformative prior over each treat_j

```
data {
  int<lower = 0> J;
  int hz[J];
  int S_N[J];
  int E_N[J];
  vector[J] S_mean;
  vector[J] E_mean;
  vector[J] S_se;
  vector[J] E_se;
}
parameters {
  real treat_effect[J];
  real a;
}
transformed parameters {
  vector[J] sigma;
  vector[J] std;

  for (j in 1:J) {
    std[j] <- (sqrt(E_N[j]) * E_se[j]);
    sigma[j] <- pow(std[j], 2);
  }
}
model {
  for (j in 1:J) {
    E_mean[j] ~ normal(treat_effect[j], sigma[j]);
  }
}
```

```r
library ("rstan")
setwd("~/Documents/BDA/Homework 3")
chickens <- read.table("chickens_data.txt", header=TRUE)
hz <- chickens$Hz
S_N <- chickens$S_N
E_N <- chickens$E_N
S_mean <- chickens$S_Mean
E_mean <- chickens$E_Mean
S_se <- chickens$S_SE
E_se <- chickens$E_SE
J <- length(hz)

fit1 <- stan("chickens1.stan")
```

From this we get that the treatment effect at each level of electromagnetic activity was fairly similar.

```r
plot(fit1, pars = "treat_effect")
```



B. So we then tested a model that put a prior on the treat effects to see if they were generated from the same distribution. We used $\text{treat}_j \sim \mathcal{N}(1, 1)$

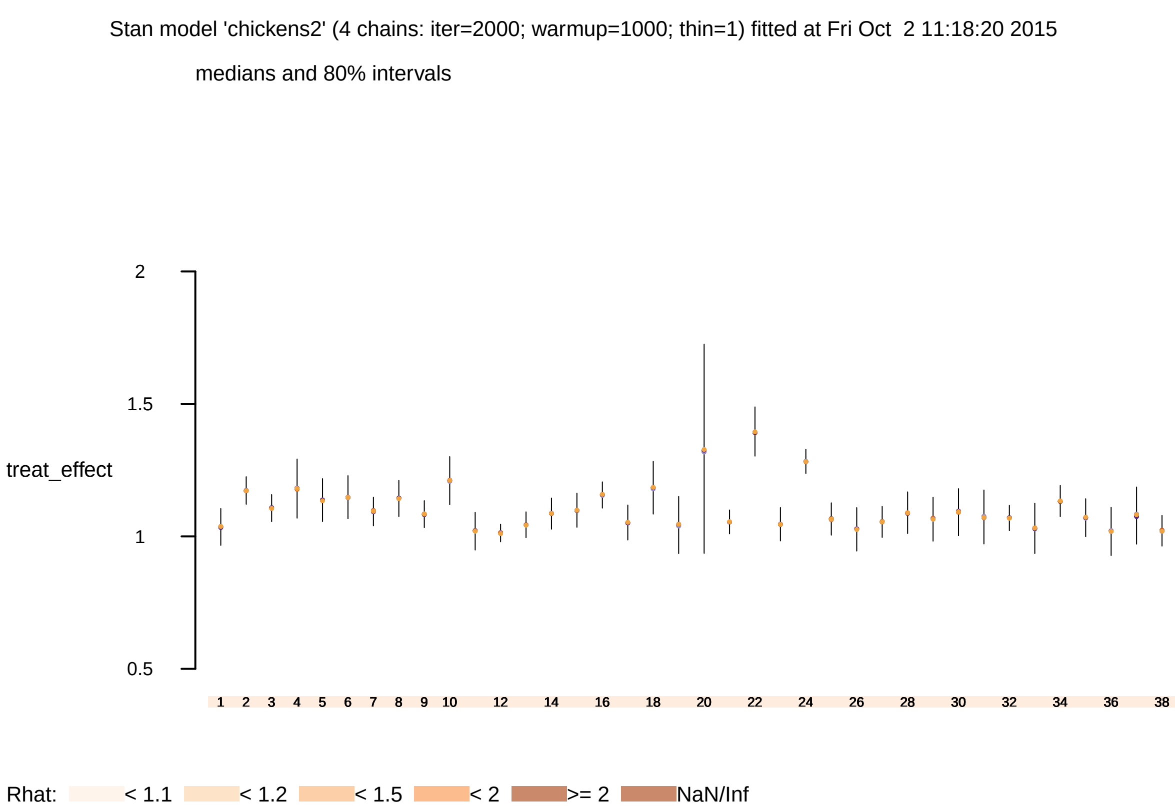
```
data {
  int<lower = 0> J;
  int hz[J];
  int S_N[J];
  int E_N[J];
  vector[J] S_mean;
  vector[J] E_mean;
```

```
  vector[J] S_se;
  vector[J] E_se;
}
parameters {
  real treat_effect[J];
}
transformed parameters {
  vector[J] sigma;
  vector[J] std;

  for (j in 1:J) {
    std[j] <- (sqrt(E_N[j]) * E_se[j]);
    sigma[j] <- pow(std[j], 2);
  }
}
model {
  for (j in 1:J) {
    treat_effect[j] ~ normal(1, 1);
    E_mean[j] ~ normal(treat_effect[j], sigma[j]);
  }
}
```

```
fit2 <- stan("chickens2.stan")
```

This didn't seem to affect the fit much at all suggesting that the treatment effects were all drawn from a similar distribution

```
plot(fit2, pars = "treat_effect")
```



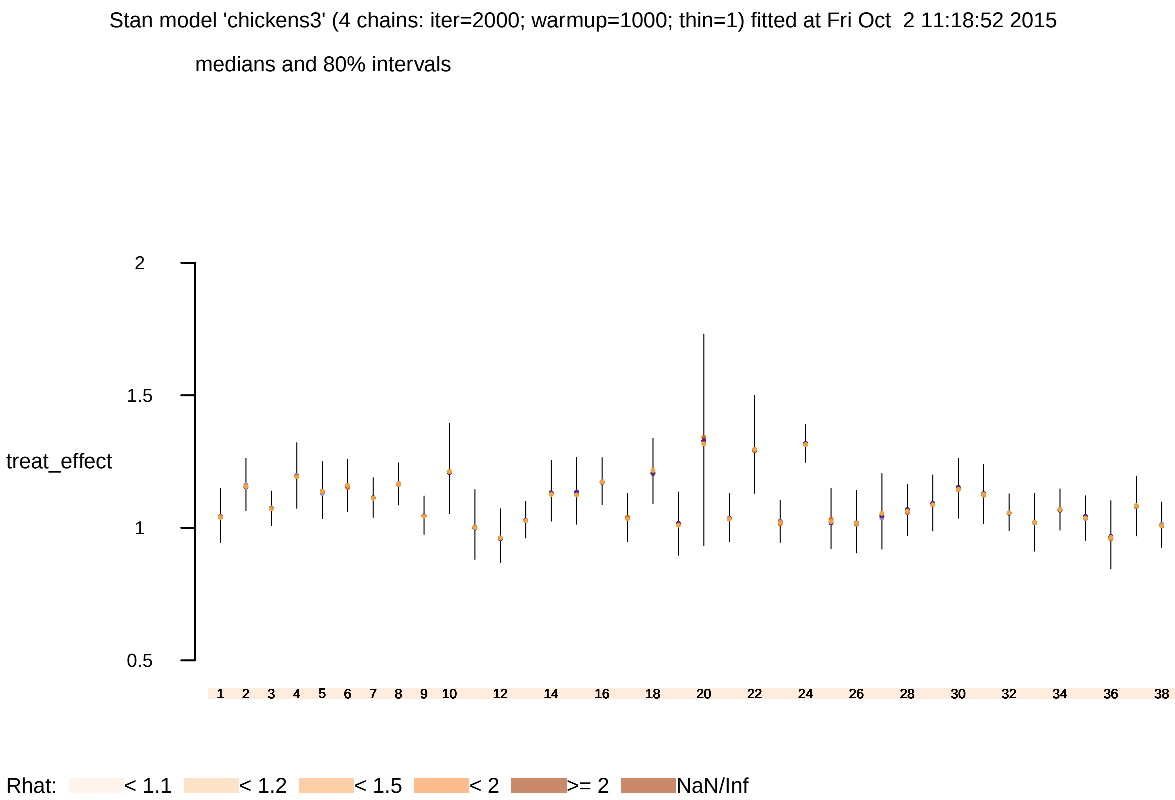
C.

The sham treatment ensures that the effect is not simply due to being in air rather than water. D. We fit a third model where we modeled the treatment effects for both the sham and the shocked chicken brains (using the same model). This actually improved the fit, going from a log posterior of 90 to 186.

```
data{
  int<lower = 0> J;
  int hz[J];
  int S_N[J];
  int E_N[J];
  vector[J] S_mean;
  vector[J] E_mean;
  vector[J] S_se;
  vector[J] E_se;
}
parameters {
  real treat_effect_E[J];
  real treat_effect_S[J];
}
transformed parameters {
  vector[J] sigma_E;
  vector[J] sigma_S;
  vector[J] std_E;
  vector[J] std_S;

  for (j in 1:J) {
    std_E[j] <- (sqrt(E_N[j]) * E_se[j]);
    sigma_E[j] <- pow(std_E[j], 2);
    std_S[j] <- (sqrt(S_N[j]) * S_se[j]);
    sigma_S[j] <- pow(std_S[j], 2);
  }
}
model {
  for (j in 1:J) {
    treat_effect_E[j] ~ normal(1, 1);
    treat_effect_S[j] ~ normal(1, 1);
    E_mean[j] ~ normal(treat_effect_E[j], sigma_E[j]);
    S_mean[j] ~ normal(treat_effect_S[j], sigma_S[j]);
  }
}
generated quantities {
  real treat_effect[J];
  for (j in 1:J)
    treat_effect[j] <- treat_effect_E[j]/treat_effect_S[j];
}
```

```r
fit3 <- stan("chickens3.stan")
```

```r
plot(fit3, pars = "treat_effect")
```



Stan model 'chickens3' (4 chains: iter=2000; warmup=1000; thin=1) fitted at Fri Oct 2 11:18:52 2015
medians and 80% intervals
treat_effect
2
1.5
1
0.5
1 2 3 4 5 6 7 8 9 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38
Rhat:
< 1.1
< 1.2
< 1.5
< 2
>= 2
NaN/Inf