STAT 491 - Lecture 9: T-tests

Posterior Predictive Distribution

Another valuable tool in Bayesian statistics is the posterior predictive distribution.

- The posterior predictive distribution can be written as:

$$p(y^*|y) = \int p(y^*|\theta)p(\theta|y)d\theta$$

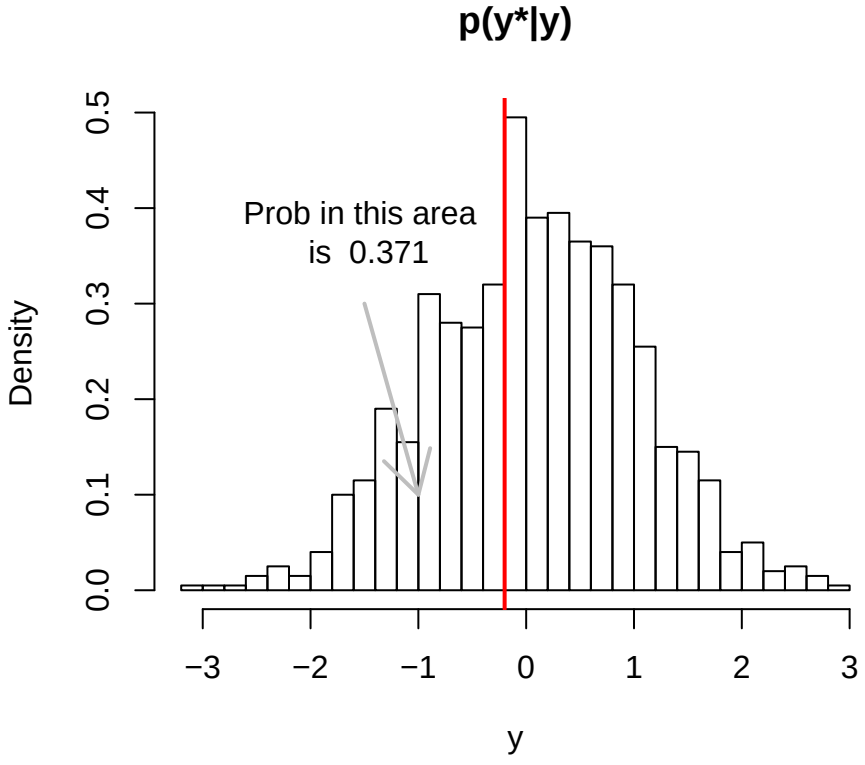
where y^* is interpreted as a new observation and $p(\theta|y)$ is the posterior for the parameter θ given that data y have been observed.

- The posterior predictive distribution allows us to test whether our sampling model and observed data are reasonable. We will talk more about this later.

- The posterior predictive distribution can also be used to make probabilistic statements about the next response, rather than the group mean. In our continuing example, we could calculate the probability of the next observed data point being greater than -0.2.

- When $p(\theta|y)$ does not have a standard form, samples from this distribution can be inserted into the sampling model. This sampling procedure is a Monte Carlo approach for this integration.

```
posterior.mu <- codaSamples[[1]][,'mu']
posterior.sigma <- codaSamples[[1]][,'sigma']
posterior.pred <- rnorm(num.mcmc, mean = posterior.mu, sd = posterior.sigma)
prob.greater <- mean(posterior.pred > -0.2)
```



T - distribution

- While the normal distribution is often used for modeling continuous data, an alternative is the t-distribution. *Q:* Where have you seen a t-distribution before and what properties does it have?

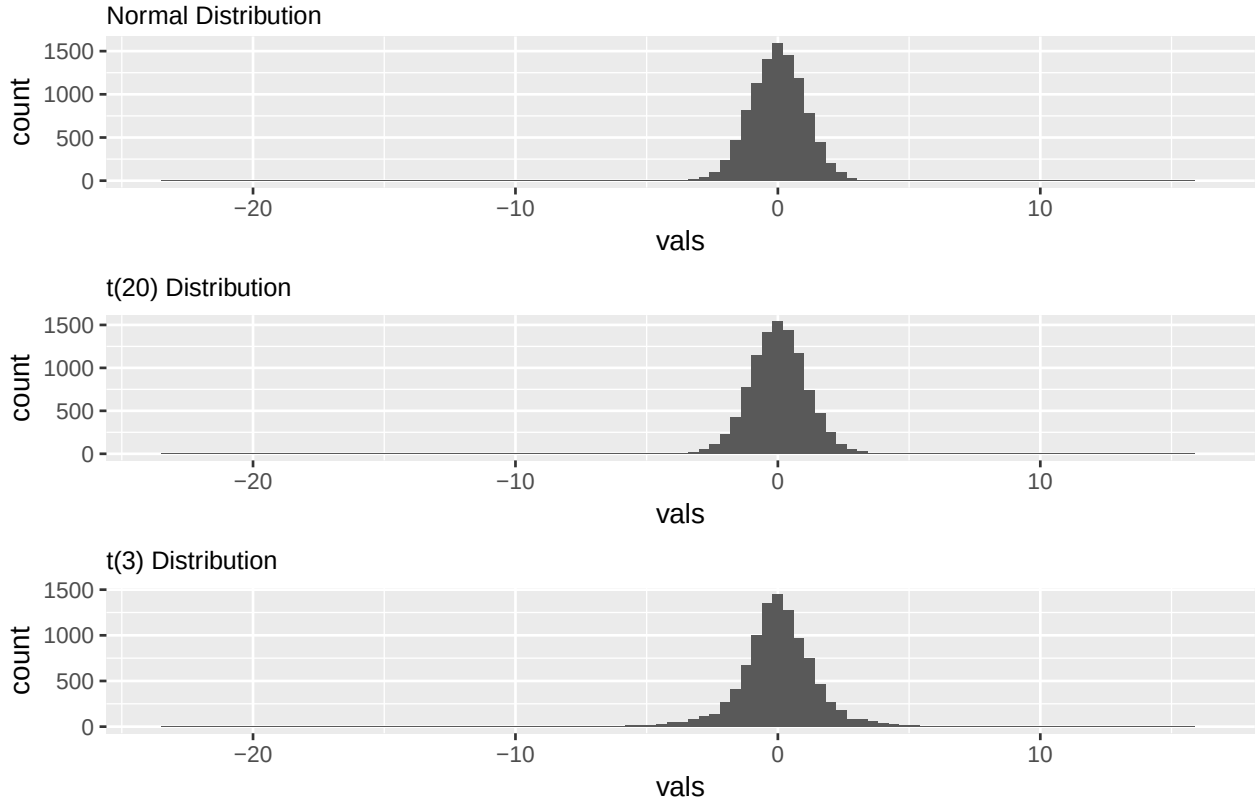
- The t-distribution has an interesting history.

- The t-distribution is

- The

  - normal = 1.96
  - t(50) =
  - t(40) =
  - t(30) =
  - t(20) =
  - t(10) =
  - t(5) =
  - t(3) =
  - t(1)

- When the degrees of freedom get large



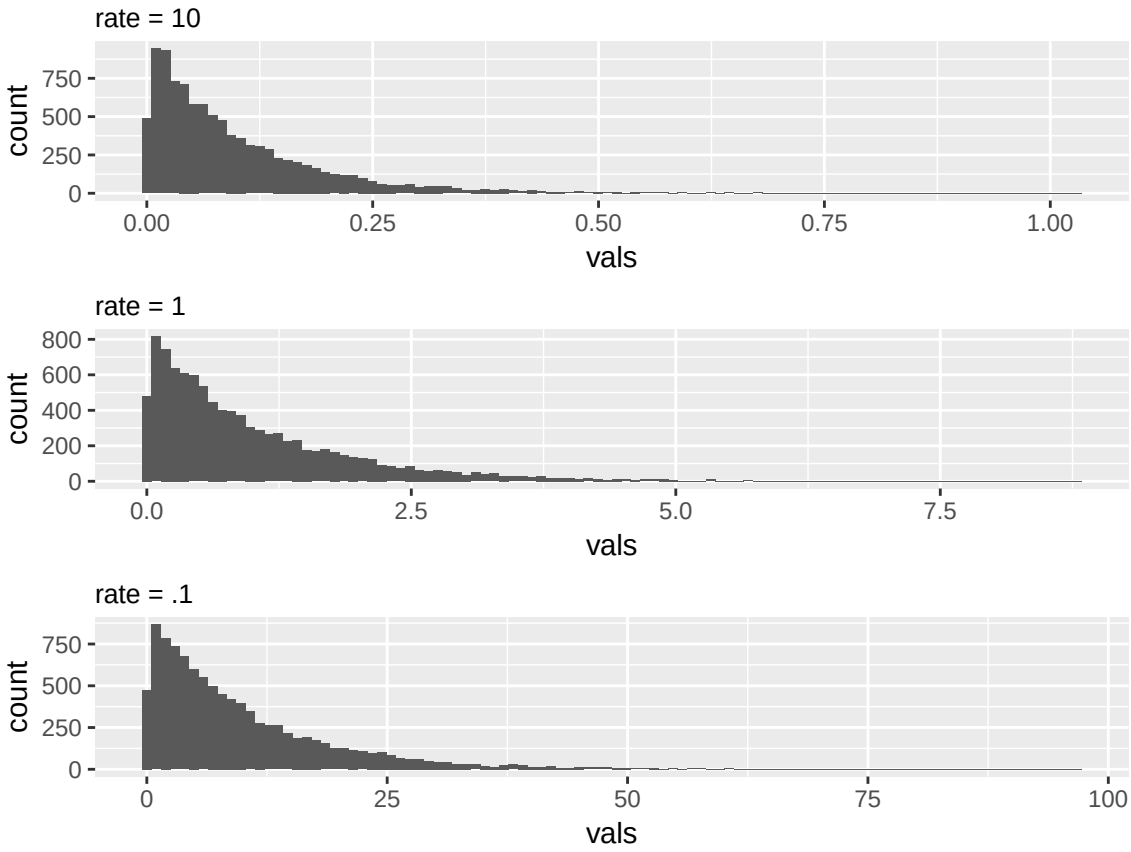
Bayesian modeling with t-distribution

- Sampling model $y \sim t(\mu, \sigma^2, \nu)$
- This requires a prior distribution on:

–

–

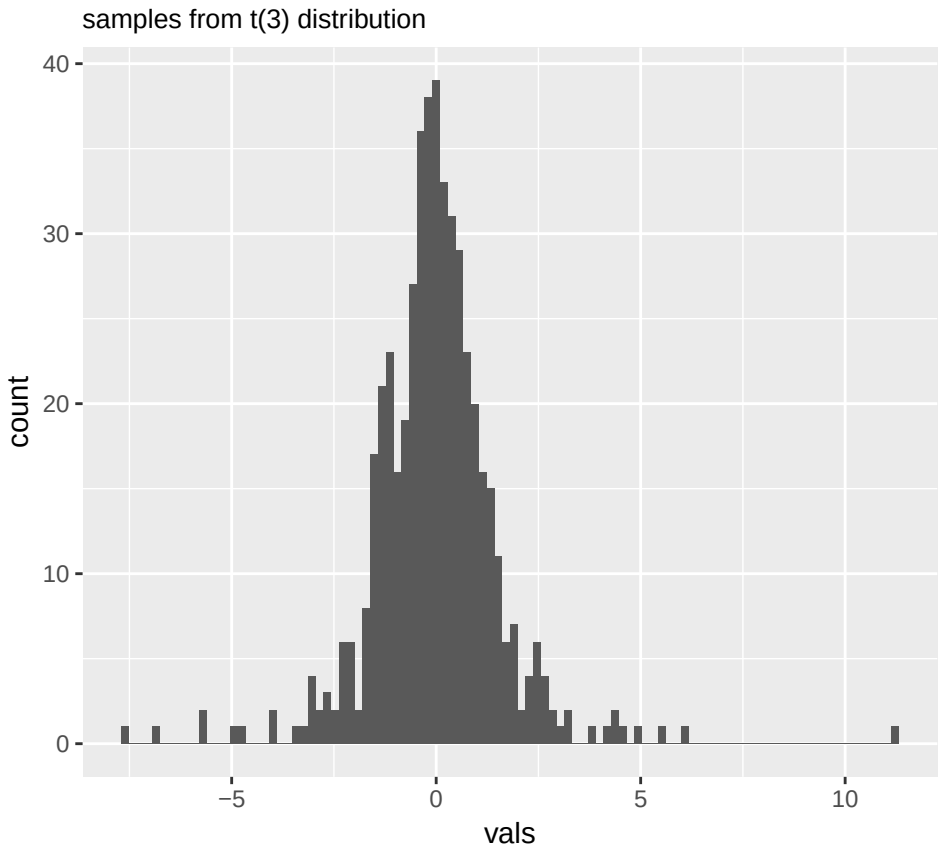
–



After looking at the figures, what do you think the mean of the exponential distribution is?

JAGS code

```
t.samples <- data.frame(rt(500, df = 3))
colnames(t.samples) <- 'vals'
ggplot(data=t.samples, aes(vals)) + geom_histogram(bins = 100) +
  labs(subtitle = "samples from t(3) distribution")
```

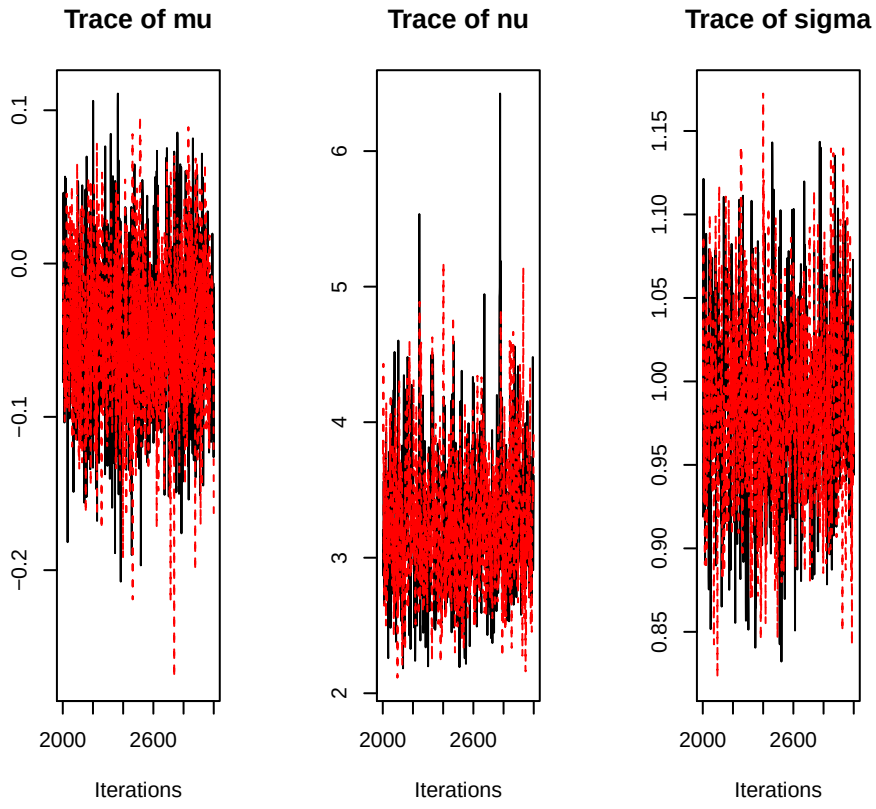



```
#Prior parameters
M <- 0
S <- 100
C <- 10
rate <- .1

# Store data
dataList = list(y = t.samples$vals, Ntotal = nrow(t.samples), M = M, S = S, C = C, rate = rate)

# Model String
modelString = "model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dt(mu, 1/sigma^2, nu) # sampling model
  }
  mu ~ dnorm(M,1/S^2)
  sigma ~ dunif(0,C)
  nu <- nuMinusOne + 1 # transform to guarantee n >= 1
  nuMinusOne ~ dexp(rate)
} "
writeLines( modelString, con='Tmodel.txt')

# initialization
initsList <- function(){
  # function for initializing starting place of theta
  # RETURNS: list with random start point for theta
  return(list(mu = rnorm(1, mean = M, sd = S), sigma = runif(1,0,C),
```

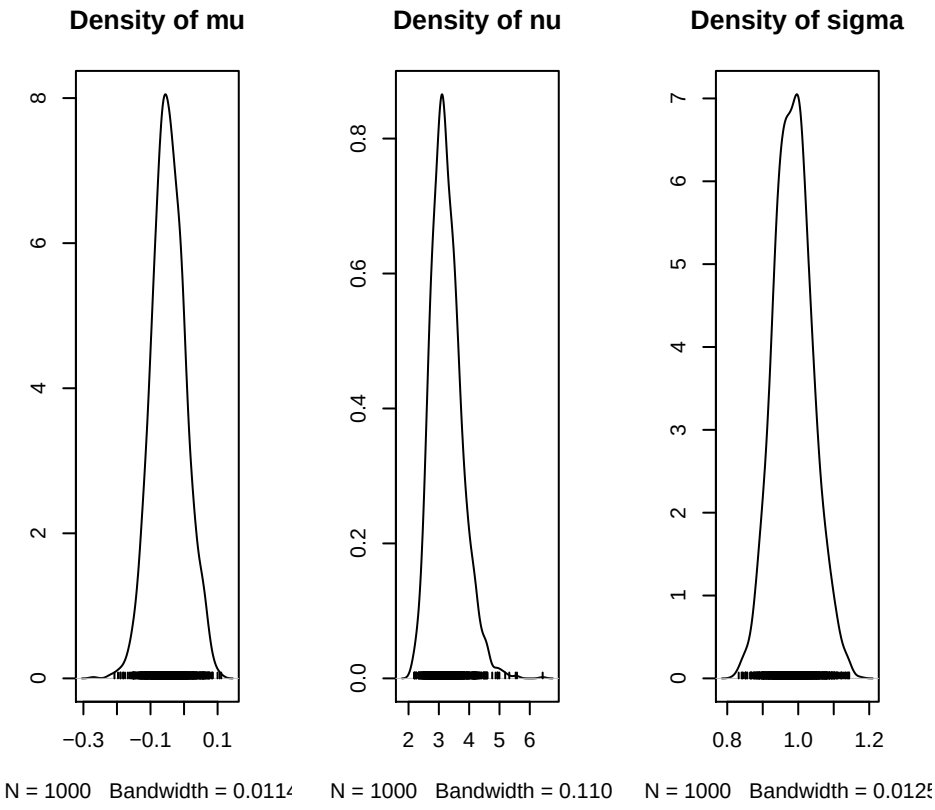
```
              nuMinusOne = rexp(1, rate=rate) ))
}

# Runs JAGS Model
jagsT <- jags.model( file = "Tmodel.txt", data = dataList, inits =initsList,
                     n.chains = 2, n.adapt = 1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 500
##    Unobserved stochastic nodes: 3
##    Total graph size: 516
##
## Initializing model
```

```
update(jagsT, n.iter = 1000)

num.mcmc <- 1000
codaSamples <- coda.samples( jagsT, variable.names = c('mu', 'sigma','nu'), n.iter = num.mcmc)

par(mfcol=c(1,3))
traceplot(codaSamples)
```




```
densplot(codaSamples)
```




```
HPDinterval(codaSamples)
```

```
## [[1]]
##            lower      upper
## mu    -0.1345267 0.06481338
## nu     2.3706040 4.25671205
## sigma  0.8815956 1.09601229
## attr(,"Probability")
## [1] 0.95
##
## [[2]]
##            lower      upper
## mu    -0.1362145 0.05478227
## nu     2.3013809 4.20014294
## sigma  0.8803749 1.08962612
## attr(,"Probability")
## [1] 0.95
```

Lab Exercise 1

1.

Simulate 100 responses from a Cauchy distribution, t distribution with $\mu = 1$, σ^2=1 and $\nu = 1$, and describe this data with a plot and brief description of the data.

2.

Use JAGS to fit a normal sampling model and the following priors for this data.

- $p(\mu) \sim N(0, 10^2)$
- $p(\sigma) \sim U(0, 1000)$

Discuss the posterior HDIs for μ and σ.

3.

Use JAGS to fit a t sampling model and the following priors for this data.

- $p(\mu) \sim N(0, 10^2)$
- $p(\sigma) \sim U(0, 1000)$
- $p(\nu) \sim E_+(.1)$, where $E_+(.1)$ is a shifted exponential with rate = .1.

Discuss the posterior HDIs for μ, σ, and ν.

4.

Use the following code to create posterior predictive distributions for part 2 and part 3. Note: your data and coda objects may need to be renamed for this to work. Compare the data and the posterior predictive model curves with posterior predictive models. Note this is the final step in Bayesian data analysis: verifying that our model / prior selection is an accurate representation of the data.

```
# Posterior Predictive Normal
post.pred.normal <- rnorm(num.mcmc, coda.norm[[1]][,'mu'], coda.norm[[1]][,'sigma'] )
# Posterior Predictive t
post.pred.t <- rt(num.mcmc, df = coda.t[[1]][,'nu']) * coda.t[[1]][,'sigma'] + coda.t[[1]][,'mu']
data.comb <- data.frame(vals = c(t.samples$vals, post.pred.normal, post.pred.t),
                        model = c(rep('data',100), rep('normal', num.mcmc), rep('t',num.mcmc)))

ggplot(data.comb, aes(vals, ..density.., colour = model)) + geom_freqpoly() +
  ggtitle('Comparison of Posterior Predictive Distributions')
```

Estimation with Two Groups

A common use of the t-distribution is to make comparisons between two groups. For instance, we may be interested to determine if the mean height of two groups of OK Cupid users are different.

We can write this model out as

From a Bayesian perspective, this model will require priors on:

-
-
-

An aside on Null Hypothesis Significance Testing (NHST) (Ch. 11)

- What is the purpose of NHST?

- For instance, consider estimating whether a die has a fair probability of rolling a 6 ($\theta = 1/6$).
  - Then if we roll the die several times
  - If the actual number is far greater or less than our expectation,
  - To do this, we compute the exact probabilities of getting all outcomes.
  - The null hypothesis is commonly rejected

- It is important to note that calculating the probability of all outcomes requires both the sampling and testing procedure.

- We are not going to get into the details, but section 11.1 in the texbook details a situation where a coin is flipped 24 times and results in 7 heads. The goal is determine if the coin is fair. Depending on the sampling procedure used, the p-value can range from .017 to .103 with this dataset.

Bayesian Approach to Testing a Point Hypothesis

Consider the die rolling example. What value for (θ) would be says is meaningfully different than $\theta = 1/6 = 0.167$?

- This range around the specified value is

- Given

- A parameter value is declared to be accepted for practical purposes if that value's

- When the HDI and ROPE overlap, with the ROPE not completely containing the HDI, then neither of the above rules is satisfied and we withhold a decision.

- Note that the NHST regime provides no way to confirm a theory, rather just the ability to reject the null hypothesis.

Lab Questions

Use the OK Cupid dataset and test the following claim, the mean height OK Cupid respondents reporting their body type as athletic is different than 70.5 inches (this value is arbitrary, but is approximately the mean height of all men in the sample). Interpret the results for each scenario.

```
okc <- read.csv('http://www.math.montana.edu/ahoegh/teaching/stat408/datasets/OKCupid_profiles_clean.cs
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
okc.athletic <- okc %>% filter(body_type == 'athletic' & sex == 'm')
okc %>% filter(sex == 'm') %>% summarise(mean(height))
```

```
##   mean(height)
## 1     70.44497
```

1. Use `t.test()` with a two-sided procedure.
2. Fit a Bayesian model for μ with a ROPE of $\pm$ .5 inch. Use the following priors: $p(\mu) \sim N(70.5, 10^2)$, $p(\sigma) \sim Unif(0, 20)$, $p(\nu) \sim E_+(.1)$ and a t-sampling model.
3. Fit a Bayesian model for μ with a ROPE of $\pm$ .05 inch

Back to the two-sample case

- Now consider whether their is a significant height difference between male OK Cupid respondents self-reporting their body type as "athletic" and those self-reporting their body type as "fit"
- From the frequentist paradigm, this can be accomplished using the `t.test()` function.

```
okc.fit <- okc %>% filter(sex == 'm' & body_type == 'fit')
t.test(x= okc.athletic$height, y = okc.fit$height)
```

```
##
##  Welch Two Sample t-test
##
## data:  okc.athletic$height and okc.fit$height
## t = 4.5651, df = 6878.8, p-value = 5.08e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1771015 0.4436713
## sample estimates:
## mean of x mean of y
##  70.68023  70.36985
```

- It is important to note there is no analog to ROPE with the t-test.
- Here is the Bayesian attempt, using JAGS. We want the posterior of $\mu_{ath} - \mu_{fit}$ for inferences.

```
M <- 70.5
S <- 10
C <- 20
rate <- .1

# Store data
dataList = list(x = okc.athletic$height, y = okc.fit$height, M = M, S = S, C = C, rate = rate)

# Model String
modelString <- "model {
  for(i in 1:length(x)) {
    x[i] ~ dt( mu_x , 1/sigma_x^2 , nu )

  }
  x_pred ~ dt( mu_x , 1/sigma_x^2 , nu ) # posterior predictive for x

  for(i in 1:length(y)) {
    y[i] ~ dt( mu_y , 1/sigma_y^2 , nu )
  }
  y_pred ~ dt( mu_y , 1/sigma_y^2 , nu ) # posterior predictive for y

  mu_diff <- mu_x - mu_y

  # The priors
  mu_x ~ dnorm( M , 1/S^2 )
  sigma_x ~ dunif( 0 , C )

  mu_y ~ dnorm( M , 1/S^2  )
  sigma_y ~ dunif( 0 , C )

  nu <- nuMinusOne+1
```

```
  nuMinusOne ~ dexp(rate)
}"
writeLines( modelString, con='TwoSampleT.txt')

# initialization
initsList <- function(){
  # function for initializing starting place of theta
  # RETURNS: list with random start point for theta
  return(list(mu_x = rnorm(1, mean = M, sd = S), sigma_x = runif(1,0,C),
              mu_y = rnorm(1, mean = M, sd = S), sigma_y = runif(1,0,C),
              nuMinusOne = rexp(1, rate=rate) ))
}

# Runs JAGS Model
jagsT <- jags.model( file = "TwoSampleT.txt", data = dataList, inits =initsList,
                     n.chains = 3, n.adapt = 1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 7034
##    Unobserved stochastic nodes: 7
##    Total graph size: 7056
##
## Initializing model
```

```
update(jagsT, n.iter = 1000)

coda.t <- coda.samples( jagsT, variable.names = c('mu_x', 'sigma_x','nu', 'mu_y', 'sigma_y', 'mu_diff')

HPDinterval(coda.t)
```

```
## [[1]]
##              lower      upper
## mu_diff  0.1997238  0.4664428
## mu_x    70.6009288 70.7806265
## mu_y    70.2652849 70.4654816
## nu      16.8730598 33.5477098
## sigma_x  2.6476684  2.7982698
## sigma_y  2.6392388  2.7960681
## attr(,"Probability")
## [1] 0.95
##
## [[2]]
##              lower     upper
## mu_diff  0.1868814  0.447558
## mu_x    70.5966013 70.777935
## mu_y    70.2728970 70.462940
## nu      15.7674326 34.378339
## sigma_x  2.6492174  2.798063
## sigma_y  2.6415375  2.794945
## attr(,"Probability")
## [1] 0.95
```

```
##
## [[3]]
##              lower      upper
## mu_diff  0.1944301  0.4544899
## mu_x    70.5981842 70.7747505
## mu_y    70.2687118 70.4700662
## nu      16.1324632 34.0206257
## sigma_x  2.6492247  2.7998339
## sigma_y  2.6419976  2.7982237
## attr(,"Probability")
## [1] 0.95
```

Given that the HDI for the difference in mean heights is 0.1997238, 0.4664428 the interpretation here depends on our ROPE.