STAT40850 - Assignment 2 - Intro to Bayesian Analysis

Conor Ryan - 16340116

2023-02-28

Introduction

In this assignment we explore the "KungSan" dataset which contains height (cm), weight (kg), age (years) and gender information about the Kung San people who live in the Kalahari desert. The data was collated by Nancy Howell between August 1967 and May 1969 (from TensorFlow website https://www.tensorflow.org/datasets/catalog/howell).

Below I read in the KungSan dataset and output the first five rows.

```r
data <- read.csv("kungsan.csv")

knitr::kable(head(data,5))
```

X	height	weight	age	male
1	151.765	47.82561	63	1
2	139.700	36.48581	63	0
3	136.525	31.86484	65	0
4	156.845	53.04191	41	1
5	145.415	41.27687	51	0

Question 1

In question 1 we are asked to work with a subset of the KungSan data set pertaining to individuals who are 18 years or older. We wish to use the Bayesian posterior model developed in Chapter 3 to estimate a posterior height for 4 individuals whose weights (kg) are 53.3, 35.7, 48.2 and 62.9. We will also produce the 90% credible interval for the estimated posterior heights for each of the 4 new individuals.

Firstly, I subset the data to include only individuals who are 18 years or older. I then create a centered weight column which will be used in the Bayesian model to greatly reduce the correlation between α and β in the linear regression model, making the MCMC algorithm in Stan more efficient. I also output the first five rows of this new data set.

```r
# subset overall KungSan dataset to individuals who are 18 or older
data_old <- data[data$age >= 18,]

# create a centered weight column
data_old$weight_c <- data_old$weight - mean(data_old$weight)

# display first 5 rows of subsetted KungSan dataset
knitr::kable(head(data_old,5))
```

X	height	weight	age	male	weight_c
1	151.765	47.82561	63	1	2.835121
2	139.700	36.48581	63	0	-8.504679
3	136.525	31.86484	65	0	-13.125647
4	156.845	53.04191	41	1	8.051429
5	145.415	41.27687	51	0	-3.713614

Next, I implement the m2 model developed in Chapter 3 in Stan which I will use to estimate the expected posterior height for the 4 new individuals. This model is saved as q1.Stan. This model is represented mathematically below.

$$h_i \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$

where h_i is the height of the i^{th} individual, x_i is the predictor variable, weight and α, β and σ have the below prior distributions.

$$\alpha \sim N(178, 20)$$
$$\beta \sim N(0, 10)$$
$$\sigma \sim U(0, 50)$$

To implement the model, I create a 'weight_new' dataframe to hold the weights of the new individuals and center these weights based on the average weight in the KungSan data (18+ years). I then organise the data into a list to be read by the Stan model, including the weight information of the new individuals which I will use to produce the posterior predicted heights. This is done using a generated quantities block in Stan.

```
# creating vector of weights for New individuals
weight_new <- as.data.frame(x = c(53.3,35.7,48.2,62.9))
colnames(weight_new) <- "weight"

# centering weight new using mean of original weights
weight_new$weight_c <- weight_new$weight - mean(data_old$weight)

# organising data into a list
dat1 <- list(N = nrow(data_old),
             height = data_old$height,
             weight = data_old$weight_c,
             N_new = nrow(weight_new),
             weight_new = weight_new$weight_c)

# fitting the model using Stan
fit_q1 <- stan(file = 'q1.stan',data = dat1,seed = 1759,iter = 5000)

# print output of the fitted model
print(fit_q1, probs = c(0.05, 0.5, 0.95))
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=5000; warmup=2500; thin=1;
## post-warmup draws per chain=2500, total post-warmup draws=10000.
##
```

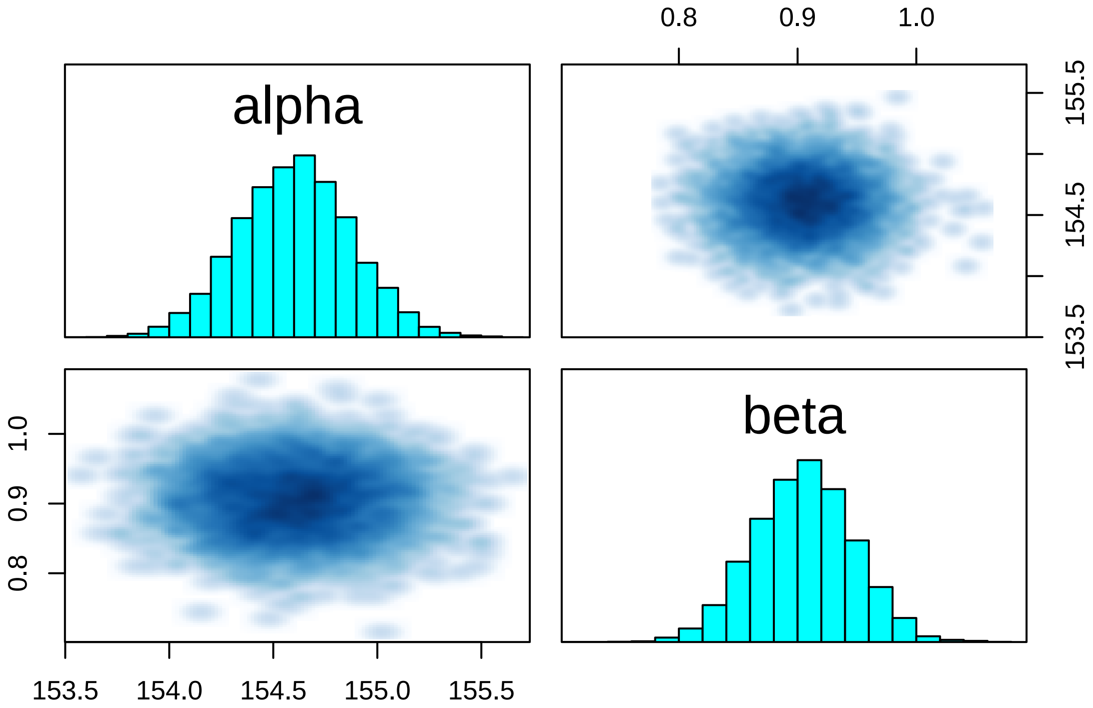
```
##                 mean se_mean   sd       5%      50%      95% n_eff Rhat
## alpha         154.60    0.00 0.27   154.16   154.61   155.06 10282    1
## beta            0.91    0.00 0.04     0.84     0.91     0.97 10812    1
## sigma           5.10    0.00 0.19     4.80     5.10     5.42 10579    1
## y_pred[1]     162.10    0.05 5.12   153.61   162.11   170.58  9919    1
## y_pred[2]     146.18    0.05 5.15   137.69   146.20   154.69  9775    1
## y_pred[3]     157.52    0.05 5.19   148.89   157.50   166.15  9719    1
## y_pred[4]     170.89    0.05 5.19   162.38   170.87   179.44 10030    1
## lp__        -1078.81    0.02 1.22 -1081.17 -1078.50 -1077.50  5280    1
##
## Samples were drawn using NUTS(diag_e) at Tue Feb 28 22:46:10 2023.
## 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).
```

As we can see from the Stan output above, the expected posterior height (cm) for each of the 4 new individuals is **162.1**,**146.18**,**157.52** and **170.89** respectively. The predicted heights of the 4 individuals correspond to the mean column of 'y_pred' rows of the output table.

The corresponding 90% credible intervals are also shown for each new individual, the lower bound of the interval is under the "5%" column and the upper bound is under the "95%" column. We can see that the 90% credible intervals are about +- 9cm wide for each of the individuals.

I also look at the pairs plot between α and β in the model to show the effect of centering the weight variable. We can see below that there is no correlation between the fitted parameters.

```
# Examining correlation between alpha and beta after centering data
#cor(cbind(post1$alpha,post1$beta))
pairs(fit_q1, pars=c("alpha", "beta"))
```



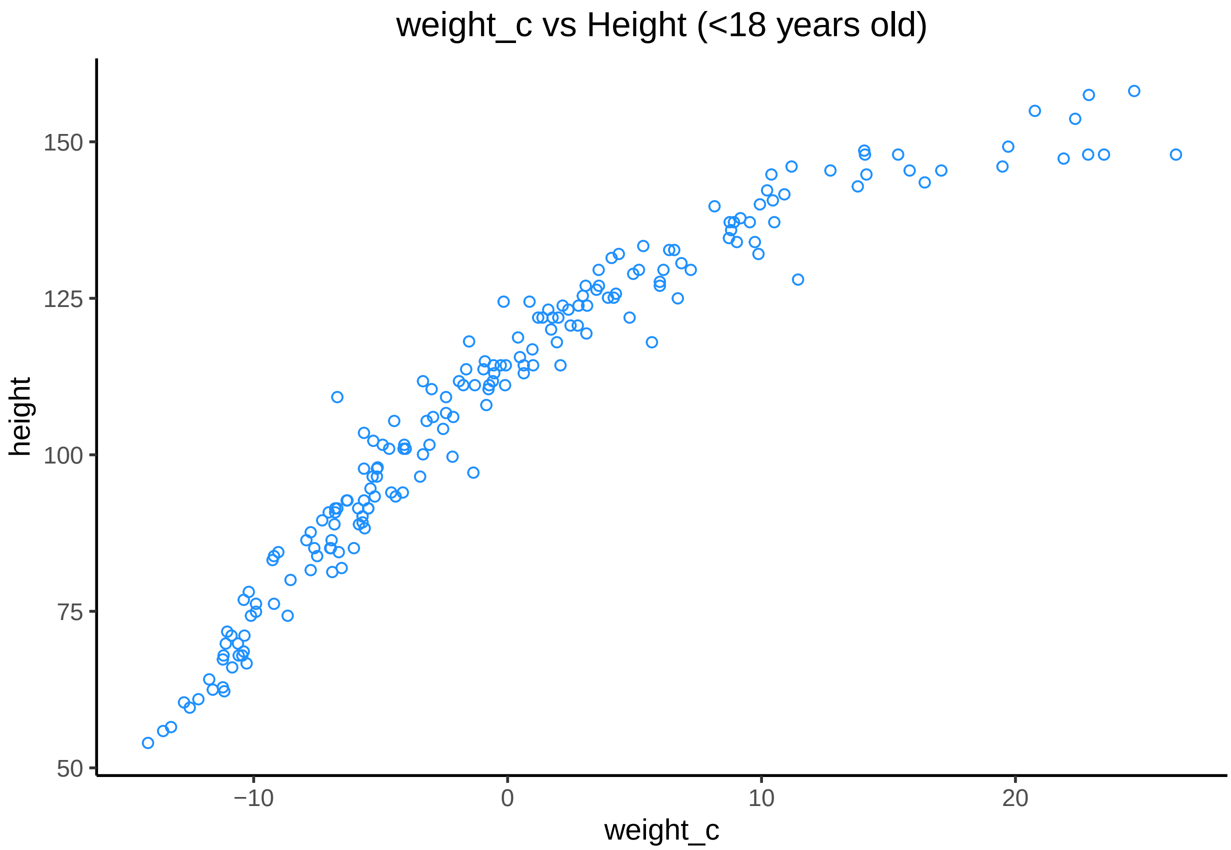
Question 2

In question 2 we are asked to explore another subset of the KungSan data set containing individuals who are less than 18 years old. I will develop a Bayesian model in order to analyse and predict the heights of these individuals based on prior assumptions and their weight.

Firstly, I subset the data to include only individuals who are less than 18 years old. I then create a centered weight variable to again reduce the correlation between fitted α and β parameters in the model. I also create a centered weight squared variable and produce a plot of the centered weight variable against height in the data.

We observe from the plot that height generally increases with weight in the data however we also note that the relationship may not be purely linear and in fact looks to be slightly quadratic.

```
# subset overall KungSan dataset to individuals who younger than 18 years
data_young <- data[data$age < 18,]

# create a centered weight column
data_young$weight_c <- (data_young$weight - mean(data_young$weight))

# create a centered weight squared column
data_young$weight_c2 <- data_young$weight_c^2

# Plot of cnetered weight variable against height
ggplot() +
  geom_point(data = data_young,
            aes(weight_c,height), shape = 1, color = 'dodgerblue') +
  theme_classic() +
  ggtitle("weight_c vs Height (<18 years old)") +
  theme(plot.title = element_text(hjust = 0.5))
```



The Bayesian model I propose to implement is similar to that of the model used in question 1 above where height (h_i) is a normally distributed variable with parameters μ and σ, where μ is distributed according to a linear regression model and σ is distributed according to a uniform prior distribution.

This model is represented mathematically below,

$$h_i \sim N(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta x_i$$

where x_i is the predictor variable weight and α, β and σ have the below prior distributions.

$$\alpha \sim N(120, 20)$$

$$\beta \sim N(0, 10)$$

$$\sigma \sim U(0, 50)$$

I have decided to reduce the mean for the prior distribution for α as we are now dealing with a subset of the KungSan dataset which contains only individuals who are younger than 18 years old. Therefore, I believe that it is more realistic to have a lower prior mean height for the intercept (α) in our linear regression.

I have kept the standard deviation of the prior distribution for α the same as I believe it sufficiently allows for a spread of alpha values about the mean.

I have used the same prior normal distribution for β as I had used in question 1. β is centered on zero which reflects the prior belief that weight has no predictive power for height however, there is a sufficiently high variance in the prior distribution for β to allow for positive (or negative) linear relationships between weight and height.

I have also used the same uniform prior distribution for σ as before.

I now Implement the above model in Stan to produce 90% credible intervals for the mean as well as 90% credible posterior predictions for predicted heights. This model is saved as q2.Stan.

```
# organising data into a list - change to centered
dat2 <- list(N = nrow(data_young),
             height = data_young$height,
             weight_c = data_young$weight_c)
```

I then fit the model using Stan, running for 5,000 iterations and setting the seed to ensure reproducible results.

```
# fitting the model using Stan
fit_q2 <- stan(file = 'q2.stan',data = dat2,seed = 1759,iter = 5000)

print(fit_q2, pars = c("alpha","beta","sigma"))
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=5000; warmup=2500; thin=1;
## post-warmup draws per chain=2500, total post-warmup draws=10000.
##
##         mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## alpha 108.32    0.01 0.61 107.11 107.91 108.31 108.74 109.53 10487    1
## beta    2.72    0.00 0.07   2.58   2.67   2.72   2.77   2.85  9821    1
## sigma   8.54    0.00 0.45   7.73   8.23   8.52   8.82   9.49  9104    1
##
## Samples were drawn using NUTS(diag_e) at Tue Feb 28 22:46:30 2023.
## 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).
```

We can see the output of our fitted Bayesian model above. The average value for the intercept α is 108.3 which represents the expected value for μ where an individuals standard weight is zero.

The average value for β the slope of the linear regression model for μ is roughly 2.7, indicating a positive relationship between weight and height.

Finally, we can see that the average value for sigma, the standard deviation for normally distributed height variable is 8.5.

Next, I extract the alpha and beta values from the output of the Stan model in order to plot the posterior mean line.

```
# extracting alpha and beta values to produce posterior mean line
post2_alpha <- extract(fit_q2)$alpha
post2_beta <- extract(fit_q2)$beta
post2 <- as.data.frame(cbind(post2_alpha,post2_beta))
colnames(post2) <- c("alpha","beta")
```

I also calculate the 90% posterior credible interval for the mean using the HDInterval package.

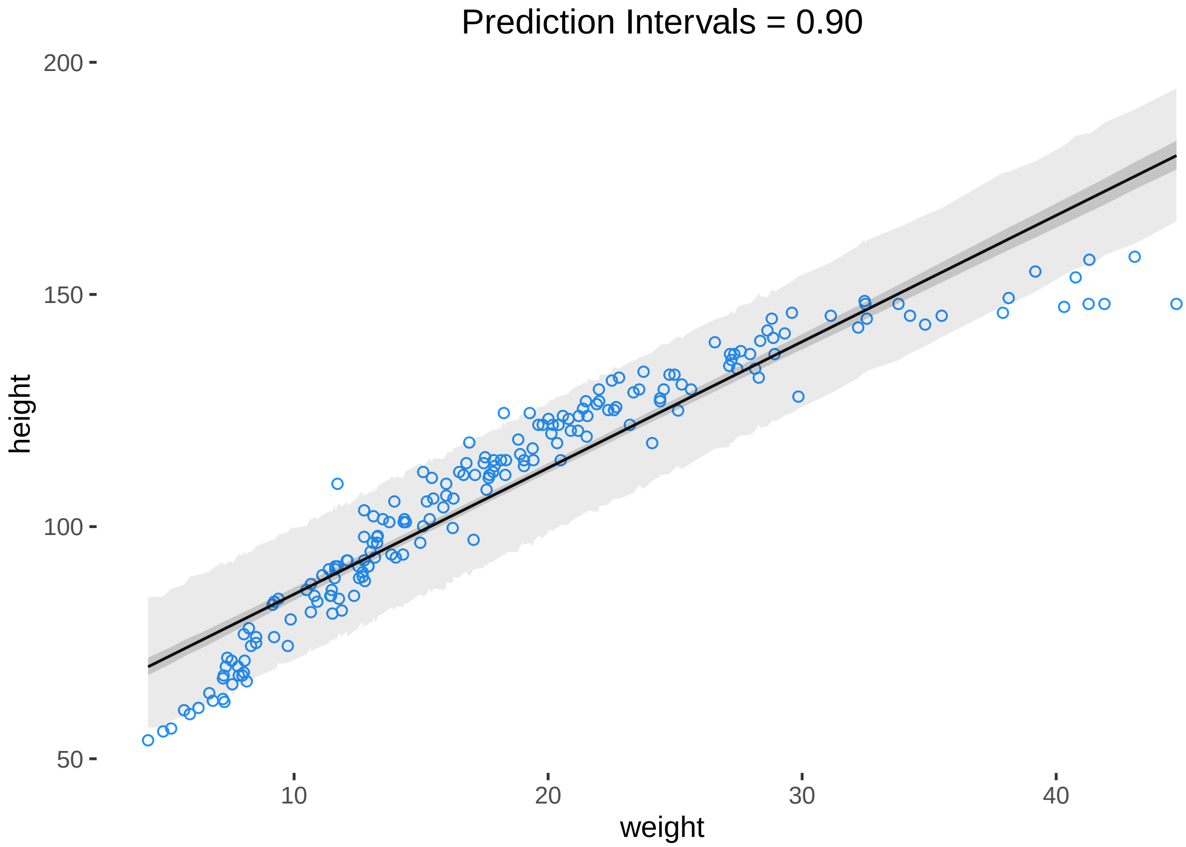
```r
# Calculating 90% posterior credible interval for the mean across each value of weight
f_mu2 <- function(x) post2$alpha + post2$beta * x
mu2 <- sapply(data_young$weight_c, f_mu2)

y_hdi2 = HDInterval::hdi(mu2, credMass=0.90)
hpdi_l2 = y_hdi2[1,]
hpdi_u2 = y_hdi2[2,]

post_y2 <- mean(post2$alpha) + mean(post2$beta)*data_young$weight_c
```

I extract the predicted heights from the model which were calculated automatically using the “generated quantities” block in Stan. These are used to produce the 90% credible interval for predicted heights.

```r
# extracting predicted from the above model to generate prediction value at each value of weight
y_pred2 <- extract(fit_q2)$y_pred
y_phdi2 <- HDInterval::hdi(y_pred2, credMass=0.90)
pi_l2 = y_phdi2[1,]
pi_u2 = y_phdi2[2,]
```

Finally, I produce a plot showing the weights against height for the individuals in the KungSan study who are less than 18 years old. I also include the posterior mean line (solid black line), the 90% credible interval for the mean (dark grey shaded area) and the 90% credible interval for the predicted heights (light grey area).

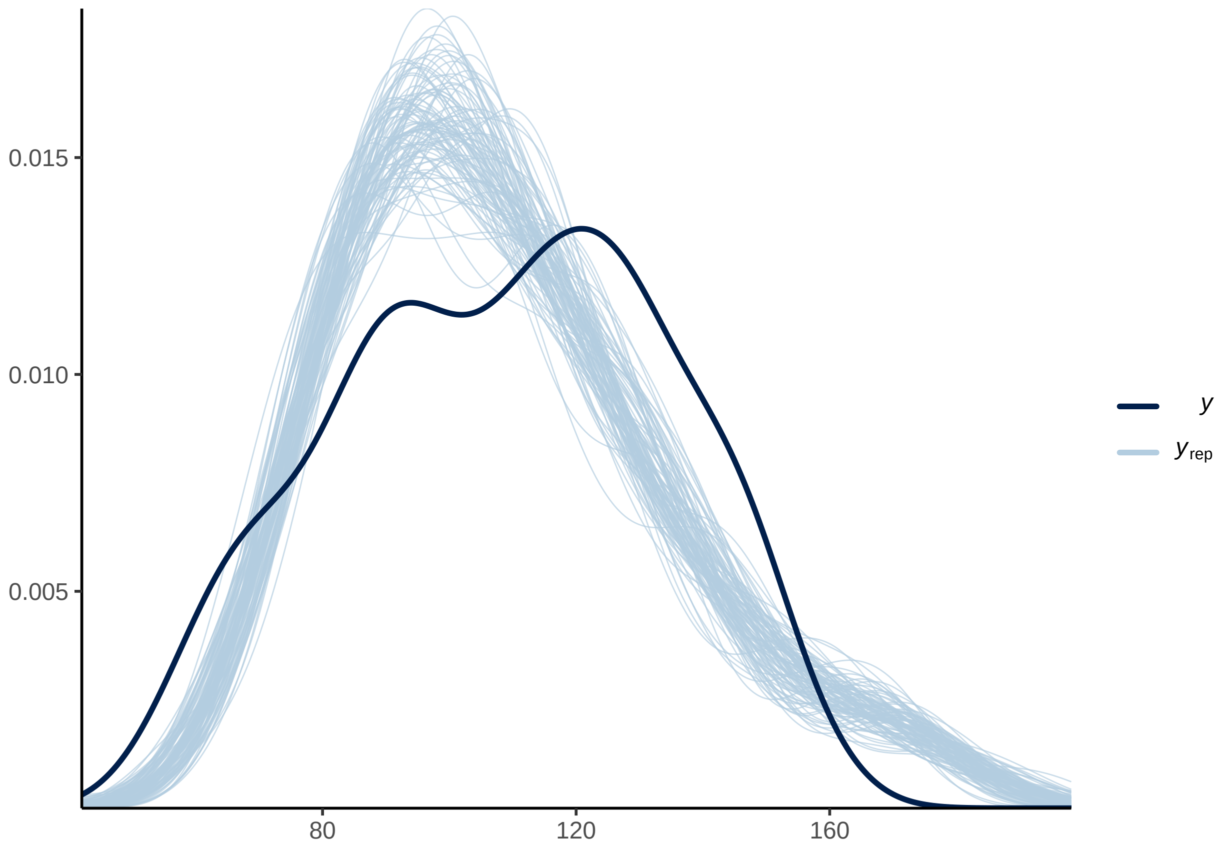
```r
# plotting 90% prediction interval for the mean and 90% credible interval for predicted heights
p <- ggplot()

p2 <- p +
  geom_point(data = data_young,
      aes(weight, height), shape = 1, color = 'dodgerblue') +
  geom_line(data = data_young,
            aes(weight,post_y2)) +
  geom_ribbon(data = data_young,
            mapping = aes(weight, ymin = hpdi_l2, ymax = hpdi_u2), alpha = .2) +
  geom_ribbon(data = data_young,
      mapping = aes(weight, ymin=pi_l2, ymax=pi_u2), alpha = .1) +
  ggtitle("Prediction Intervals = 0.90") +
  theme(plot.title = element_text(hjust = 0.5))

p2
```



As we can see from the above plot, the model does not fit the data well. There is a clear quadratic relationship present between weight and height. In the extreme weight data points (low and high) we can see we have observed weights outside of the 90% credible interval for predicted heights indicating a poor model fit. There are also areas (e.g. weights 17 - 23) where we are predicting heights above and below the posterior mean line but only observe heights above the line. This further indicates a poor model fit.

I will now use the Bayesplot package to overlay a collection of densities of replicated data sets produced from the posterior predictive distribution and compare these to the observed data.

```
# overlay 100 densities from posterior predictive over the observed dataset
ppc_dens_overlay(data_young$height, y_pred2[1:100,]) + theme_classic()
```



In the above plot we can see the observed density (dark line) and 100 overlaid simulated densities (lighter blue lines) from the posterior predictive distribution. We can see that our simulated data does not correspond well with our observed data. We can see at lower observed heights our models have less density in the low values for height, and have more density at very large heights. There is also a large difference between the observed and simulated densities in average values of heights.

The reason this model does not align well with the observed data is due to the the over-simplistic nature of the model. This can be overcome by including additional predictive variables in our model to better explain the variability in height. This could be done by adding an additional predictive variable such as the weight squared as shown in the tutorials.

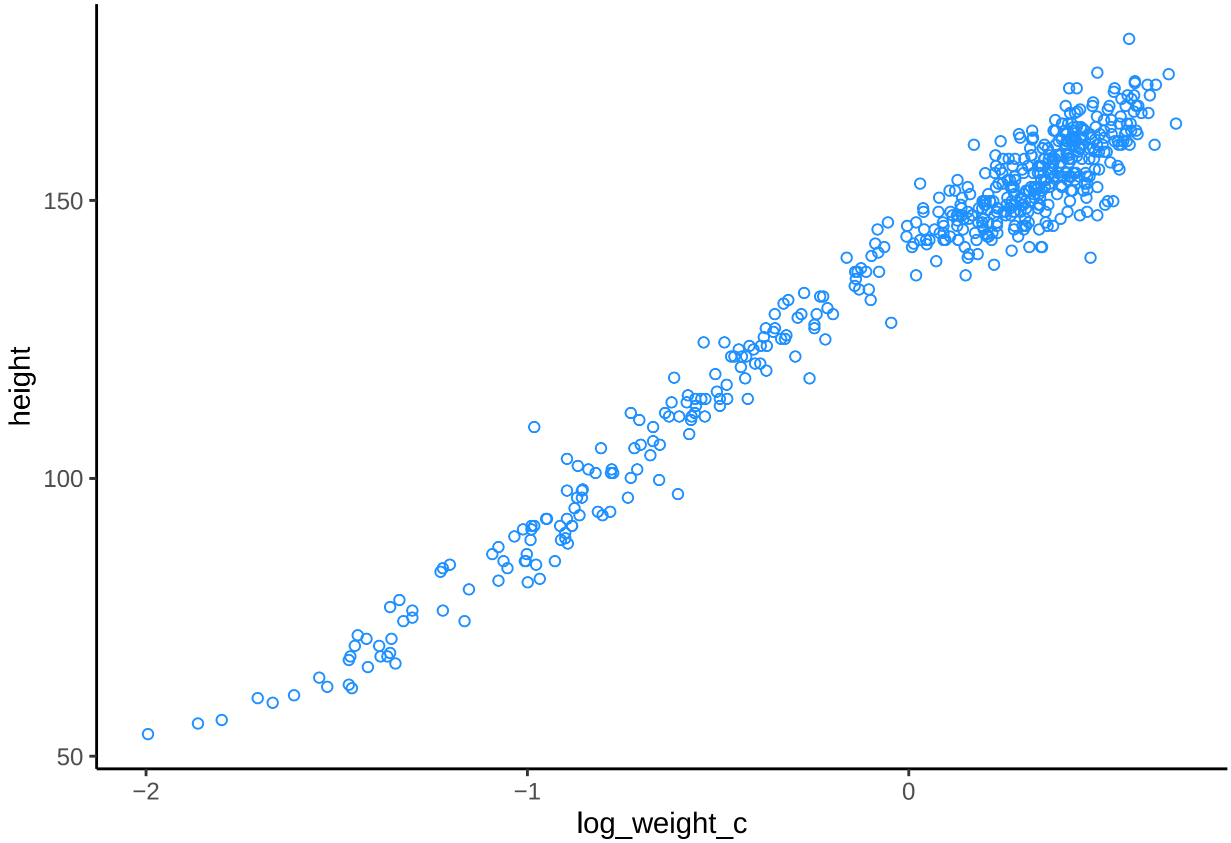
We could also add an entirely different explanatory variable to our model such as age. I believe age could work well in this particular case as all of the individuals are under the age of 18 and as such age would be a more significant factor in determining height compared to an adult population who are fully grown. The inclusion of age or weight squared as additional variables in the linear regression model for μ could be used to improve overall model fit.

I have included the overlaid 'Bayesplot' plot for each of the above mentioned models in the appendix for your reference. The inclusion of either variable improved the model fit however, weight squared resulted in the best model fit.

Question 3

In question 3 we now analyse the entire KungSan data set and wish to model the relationship between height and log(weight) for use in the posterior model. I will develop a Bayesian model in order to analyse and predict the heights of these individuals based on prior assumptions and their log(weight).

```r
# create a log weight column
data$log_weight <- log(data$weight)

# create a centered log weight column
data$log_weight_c <- (data$log_weight - mean(data$log_weight))

# Plot of log weight variable against height
ggplot() +
  geom_point(data = data,
             aes(log_weight_c,height), shape = 1, color = 'dodgerblue') +
  theme_classic()
```



For question 3, the Bayesian model I propose to implement is similar as that of the linear regression model used in question 1 above where height (y) is a Gaussian variable with parameters μ and σ and where μ is distributed according to a linear regression model and σ is distributed according to a uniform prior.

This model is represented mathematically below,

$$h_i \sim N(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta log(x_i)$$

where $log(x_i)$ is the log of weight and α, β and σ have the below prior distributions.

$$\alpha \sim N(178, 20)$$

$$\beta \sim N(0, 10)$$

$$\sigma \sim U(0, 50)$$

I have decided to use the mean of 178 for the prior alpha normal distribution as in question 3 we are considering the entire KungSan data set and not only the individuals that are less than 18 years old. I have used the same standard deviation of the prior distribution for α as used in question 2 as I believe it is sufficiently large to to allow for a spread of alpha values about the mean.

I have used the same prior normal distribution for β and σ as before.

I now Implement the above model in Stan to produce 90% credible intervals for the mean as well as 90% credible posterior predictions for predicted heights. This model is saved as q3.Stan.

Firstly, I organise the data into a list so it can be read correctly by Stan.

```
# organising data into a list - change to centered
dat3 <- list(N = nrow(data),
             height = data$height,
             log_weight_c = data$log_weight_c)
```

I then fit the model using Stan, running for 5,000 iterations and setting the seed to ensure reproducible results.

```
# fitting the model using Stan
fit_q3 <- stan(file = 'q3.stan',data = dat3,seed = 1759,iter = 5000)

#print(fit_q2)
```

Next, I extract the alpha and beta values from the output of the Stan model in order to plot the posterior mean line.

```
# extracting alpha and beta values to produce posterior mean line
post3_alpha <- extract(fit_q3)$alpha
post3_beta <- extract(fit_q3)$beta
post3 <- as.data.frame(cbind(post3_alpha,post3_beta))
colnames(post3) <- c("alpha","beta")
```

I also calculate the 90% posterior credible interval for the mean using the HDInterval package.

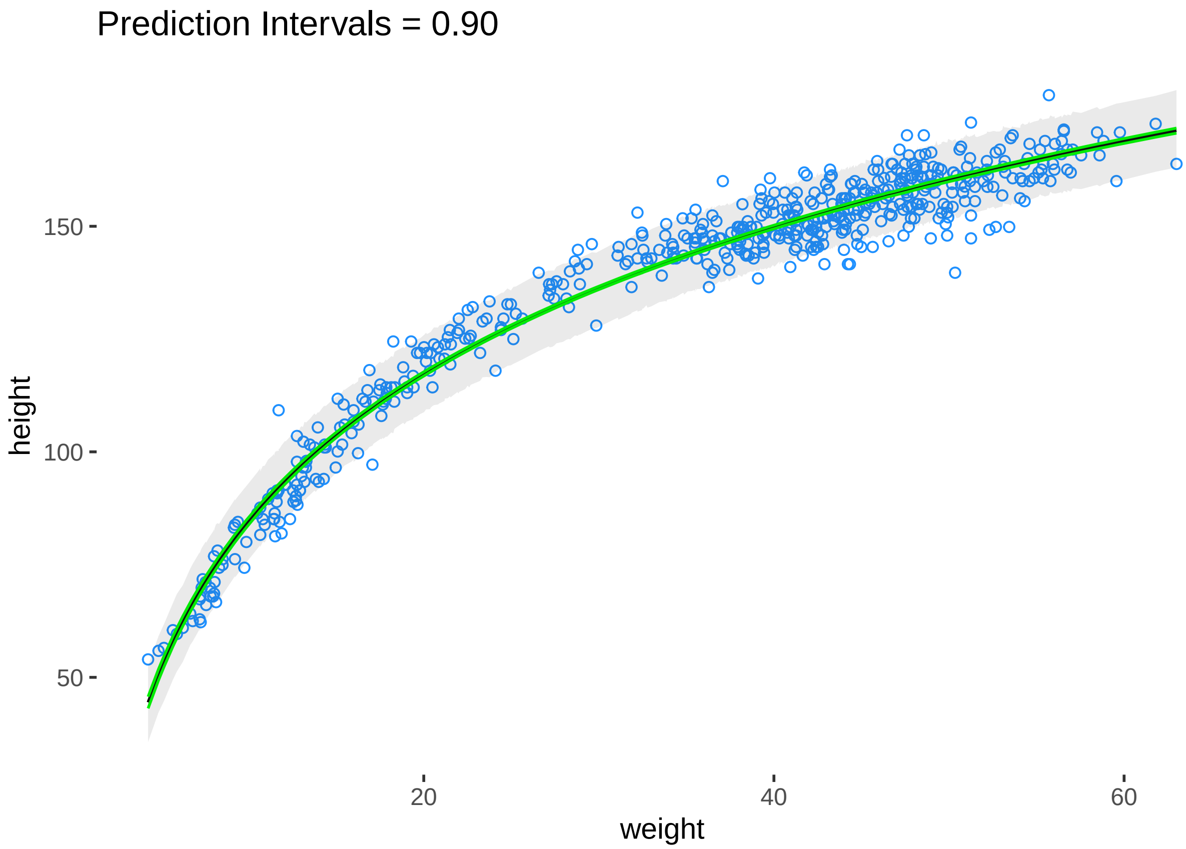
```
# Calculating 90% posterior credible interval for the mean across each value of weight
f_mu3 <- function(x) post3$alpha + post3$beta * x
mu3 <- sapply(data$log_weight_c, f_mu3)

y_hdi3 = HDInterval::hdi(mu3, credMass=0.90)
hpdi_l3 = y_hdi3[1,]
hpdi_u3 = y_hdi3[2,]

post_y3 <- mean(post3$alpha) + mean(post3$beta)*data$log_weight_c
```

I extract the predicted heights from the model which were calculated automatically using the "generated quantities" block in Stan. These are used to produce the 90% credible interval for predicted heights.

```
# extracting predicted from the above model to generate prediction value at each value of weight
y_pred3 <- extract(fit_q3)$y_pred
y_phdi3 <- HDInterval::hdi(y_pred3, credMass=0.90)
pi_l3 = y_phdi3[1,]
pi_u3 = y_phdi3[2,]
```

Finally, I produce a plot showing the weights against height for all of the individuals in the KungSan study. I also include the posterior mean line (solid black line), the 90% credible interval for the mean (dark grey shaded area) and the 90% credible interval for the predicted heights (light grey area).

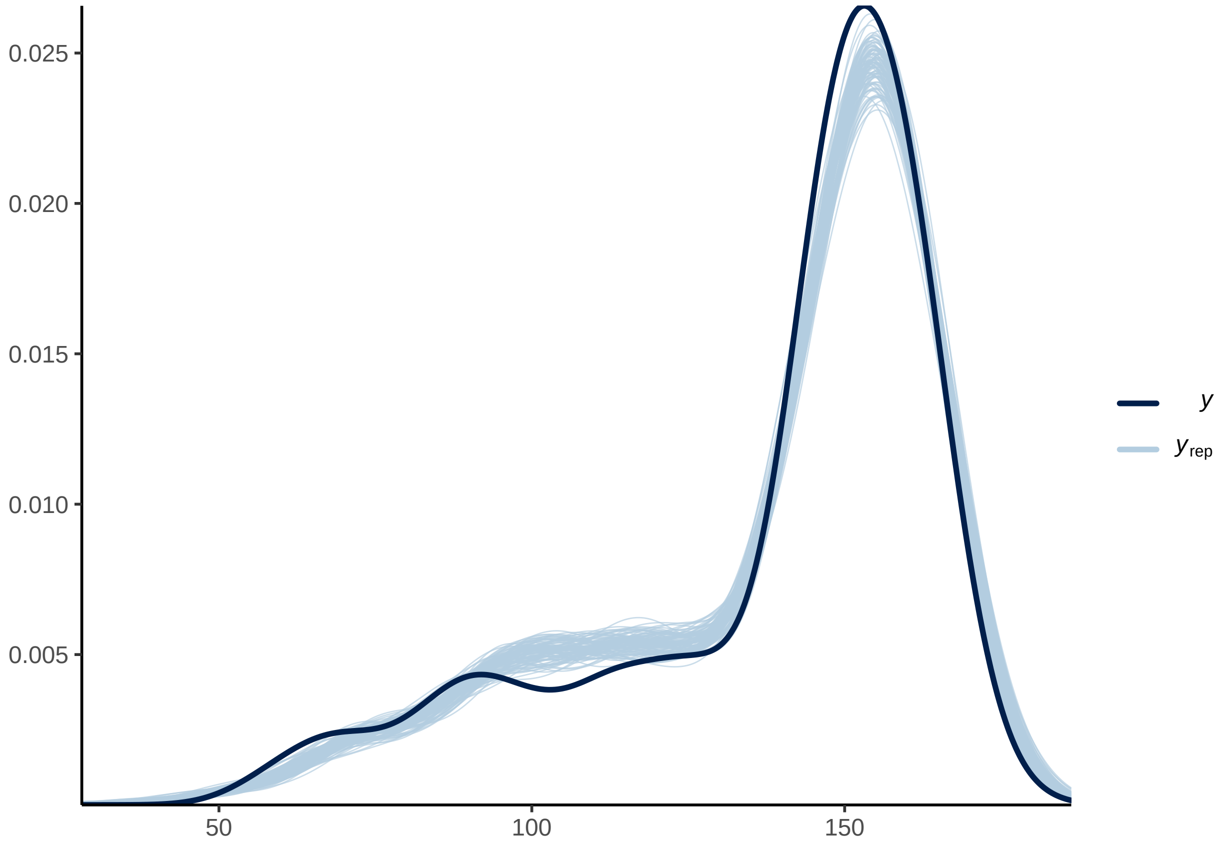
```
# plotting 90% prediction interval for the mean and 90% credible interval for predicted heights
p <- ggplot()

p3 <- p +
  geom_point(data = data,
      aes(weight, height), shape = 1, color = 'dodgerblue') +
#  geom_abline(data = post3,
#      aes(intercept = mean(alpha), slope = mean(beta))) +
  geom_line(data = data,
            aes(weight,post_y3)) +
  geom_ribbon(data = data,
      mapping = aes(weight, ymin = hpdi_l3, ymax = hpdi_u3), alpha = .2, color = 'green') +
  geom_ribbon(data = data,
      mapping = aes(weight, ymin=pi_l3, ymax=pi_u3), alpha = .1) +
  ggtitle("Prediction Intervals = 0.90")

p3
```



As we can see from the above plot, this model fits the data well. By using the log transformation of the weight variable we are able to capture the non-linear relationship between weight and height. The model is not a perfect fit as we can still see observed data points that lie outside of the 90% credible interval however the vast majority of observed data points are within the interval. We also observe very few areas where we are predicting heights above and below the mean posterior line but only observing values above (or below).

The 90% credible interval for the mean is very small in terms of variability so much so that I needed to use a green colour to distinguish the interval from the posterior mean line. To further check model fit I will now use the Bayesplot package to overlay a collection of densities of replicated data sets produced from the posterior predictive distribution and compare these to the observed data.

```
# overlay 100 densities from posterior predictive over the observed dataset
ppc_dens_overlay(data$height, y_pred3[1:100,]) + theme_classic()
```



Observing the 100 overlaid simulated densities against the true density we can see the model fits well. There are some cases where the model produced densities are larger than that of the observed particular around heights of 100 however in the main the simulated densities tracks the observed density well.

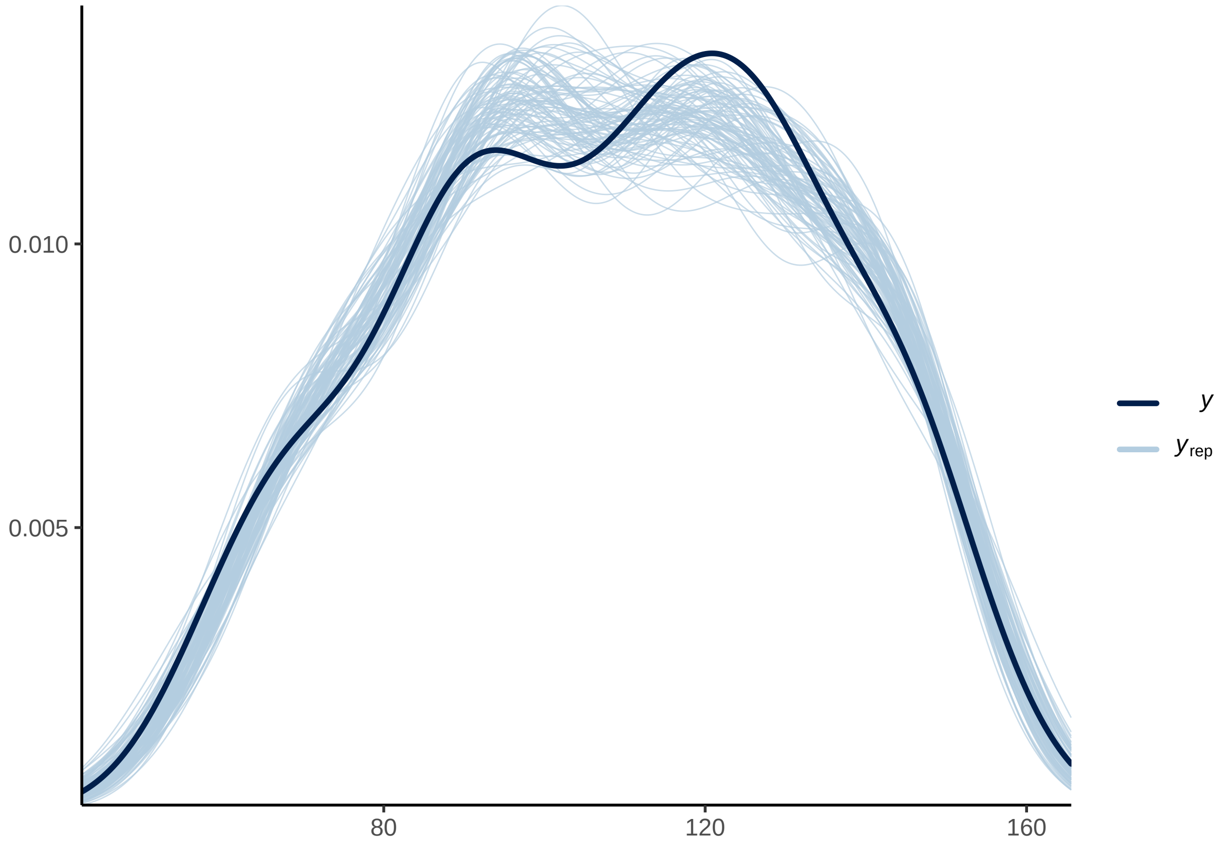
Appendix

Question 2

Model using additional squared weight variable in the linear regression model for μ.

```
# organising data into a list - change to centered
dat2_2 <- list(N = nrow(data_young),
             height = data_young$height,
             weight_c = data_young$weight_c,
             weight_c2 = data_young$weight_c2)
```

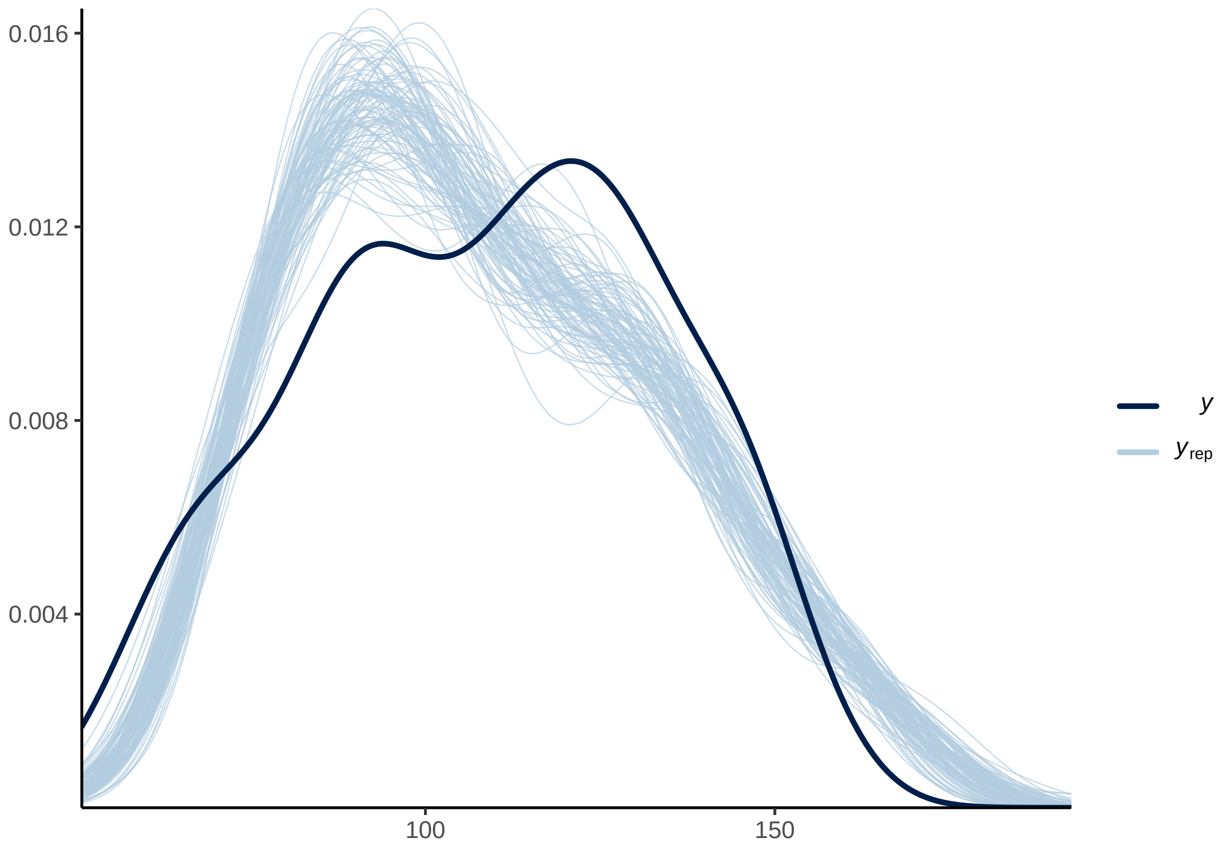
```
# fitting the model using Stan
fit_q2_2 <- stan(file = 'q2_2.stan',data = dat2_2,seed = 1759,iter = 5000)

# extracting predicted from the above model to generate prediction value at each value of weight
y_pred2_2 <- extract(fit_q2_2)$y_pred

# overlay 100 densities from posterior predictive over the observed dataset
ppc_dens_overlay(data_young$height, y_pred2_2[1:100,]) + theme_classic()
```



Model using additional age variable in the linear regression model for μ.

```
# organising data into a list - change to centered
dat2_3 <- list(N = nrow(data_young),
             height = data_young$height,
             weight_c = data_young$weight_c,
             age = data_young$age)
```

```
# fitting the model using Stan
fit_q2_3 <- stan(file = 'q2_3.stan',data = dat2_3,seed = 1759,iter = 5000)

# extracting predicted from the above model to generate prediction value at each value of weight
y_pred2_3 <- extract(fit_q2_3)$y_pred

# overlay 100 densities from posterior predictive over the observed dataset
ppc_dens_overlay(data_young$height, y_pred2_3[1:100,]) + theme_classic()
```


0.016
0.012
0.008
0.004
100
150
y
y_{rep}