Assignment2\_B04704016林家毅

# Question 1

#Create the answer sheet  
answer.sheet <- data.frame(  
 list(  
 "Individual" = c( 1 , 2 , 3 , 4 , 5 ),  
 "Weight" = c( 46.95 , 43.72 , 64.78 , 32.59 , 54.63 ),  
 "Expected height" = rep( "NA" , 5 ),   
 "89% interval" = rep( "NA" , 5 )  
 ) , check.names=FALSE  
)  
answer.sheet

## Individual Weight Expected height 89% interval  
## 1 1 46.95 NA NA  
## 2 2 43.72 NA NA  
## 3 3 64.78 NA NA  
## 4 4 32.59 NA NA  
## 5 5 54.63 NA NA

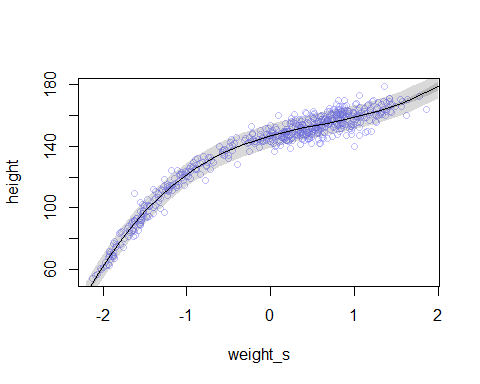
#Load data  
data(Howell1)  
Howell1.data <- Howell1  
d <- Howell1  
  
#Inspect data  
str(d)

## 'data.frame': 544 obs. of 4 variables:  
## $ height: num 152 140 137 157 145 ...  
## $ weight: num 47.8 36.5 31.9 53 41.3 ...  
## $ age : num 63 63 65 41 51 35 32 27 19 54 ...  
## $ male : int 1 0 0 1 0 1 0 1 0 1 ...

precis(d)

## mean sd 5.5% 94.5% histogram  
## height 138.2635963 27.6024476 81.108550 165.73500 ▁▁▁▁▁▁▁▂▁▇▇▅▁  
## weight 35.6106176 14.7191782 9.360721 54.50289 ▁▂▃▂▂▂▂▅▇▇▃▂▁  
## age 29.3443934 20.7468882 1.000000 66.13500 ▇▅▅▃▅▂▂▁▁  
## male 0.4724265 0.4996986 0.000000 1.00000 ▇▁▁▁▁▁▁▁▁▇

#Use cubic regression to fit the data  
#Standardize predictors  
d$weight\_s <- ( d$weight - mean(d$weight) )/sd( d$weight )  
d$weight\_s2 <- d$weight\_s^2  
d$weight\_s3 <- d$weight\_s^3  
  
#Create model using quadratic approximation  
model.fit <- quap(  
 alist(  
 height ~ dnorm( mu , sigma ) ,  
 mu <- a + b1\*weight\_s + b2\*weight\_s2 + b3\*weight\_s3 ,  
 a ~ dnorm( 178 , 20 ) ,  
 b1 ~ dlnorm( 0 , 1 ) ,  
 b2 ~ dnorm( 0 , 10 ) ,  
 b3 ~ dnorm( 0 , 10 ) ,  
 sigma ~ dunif( 0 , 50 )  
 ) ,  
 data=d )  
  
#Inspect the model  
#Create a sequence of standardized weights  
weight.seq <- seq( from=-2.2 , to=2 , length.out=30 )  
  
#Create a list of predictors  
pred\_dat <- list( weight\_s=weight.seq , weight\_s2=weight.seq^2 , weight\_s3=weight.seq^3 )  
  
#Sampling from posterior distribution, and compute mu for each value in pred\_dat  
mu <- link( model.fit , data=pred\_dat )  
  
#Calculate the mean and 89% interval for mu  
mu.mean <- apply( mu , 2 , mean )  
mu.PI <- apply( mu , 2 , PI , prob=0.89 )  
  
#Simulate 10000 predicted heights for each pred\_dat  
sim.height <- sim( model.fit , data=pred\_dat, n=1e4 )  
  
#Calculate the 89% interval of predicted heights  
height.PI <- apply( sim.height , 2 , PI , prob=0.89 )  
  
#Plot the results of model with raw data  
plot( height ~ weight\_s , d , col=col.alpha(rangi2,0.5) )  
lines( weight.seq , mu.mean )  
shade( mu.PI , weight.seq )  
shade( height.PI , weight.seq )



#The model fits quite well to the observated data, so the next step would be to predict heights from given weight data.  
  
#Standardize the weight data in answer.sheet  
weight\_predict\_s <- ( answer.sheet$Weight - mean(d$weight) )/sd( d$weight )  
  
#Create the predictors of the cube regression  
weight\_predictors <- list( weight\_s=weight\_predict\_s , weight\_s2=weight\_predict\_s^2 , weight\_s3=weight\_predict\_s^3 )  
  
#Sampling from posterior distribution, and compute mu for each weight\_predictors  
mu.predict <- link( model.fit , data=weight\_predictors )  
  
#Calculate the mean and 89% interval for mu.predict  
mu.predict.mean <- apply( mu.predict , 2 , mean )  
mu.predict.PI <- apply( mu.predict , 2 , PI , prob=0.89 )  
  
#Simulate predicted heights for each value in weight\_predictors  
sim.height.predict <- sim( model.fit , data=weight\_predictors )  
  
#Calculate the 89% interval of predicted heights  
height.predict.PI <- apply( sim.height.predict , 2 , PI , prob=0.89 )  
  
#Add mean heights into answer.sheet  
height.mean <- round( mu.predict.mean , 3 )  
answer.sheet$`Expected height` <- height.mean  
  
#Add 89% intervals of predicted heights into answer.sheet  
height.HPDI <- round( height.predict.PI , 3 )  
answer.sheet$`89% interval` <- sapply( 1:5 , function(i)  
 paste( "( " , height.HPDI[ 1 , i ] , " , " , height.HPDI[ 2 , i ] , " )" )  
)  
  
#Answer  
answer.sheet

## Individual Weight Expected height 89% interval  
## 1 1 46.95 156.072 ( 148.804 , 163.962 )  
## 2 2 43.72 153.637 ( 145.843 , 160.991 )  
## 3 3 64.78 178.679 ( 170.112 , 187.394 )  
## 4 4 32.59 143.360 ( 135.981 , 151.716 )  
## 5 5 54.63 162.951 ( 155.093 , 170.583 )

# Question2

#2-a  
#Filter out children  
d2 <- Howell1.data[ d$age < 18 , ]  
  
#Inspect date  
precis(d2)

## mean sd 5.5% 94.5% histogram  
## height 108.3188531 25.7451390 66.360675 147.63432 ▁▃▂▅▅▃▇▇▃▅▁  
## weight 18.4141931 8.9393135 7.243439 35.16429 ▁▅▇▅▅▃▁▁▁  
## age 7.7218750 5.3662354 0.303000 16.00000 ▇▃▅▃▂▃▂▃▂  
## male 0.4791667 0.5008718 0.000000 1.00000 ▇▁▁▁▁▁▁▁▁▇

#Create data list for stan model  
data\_list <- list(  
 children\_height = d2$height,  
 children\_weight = d2$weight,  
 children\_age = d2$age,  
 children\_male = d2$male  
)   
  
#Inspect data list  
str(data\_list)

## List of 4  
## $ children\_height: num [1:192] 121.9 105.4 86.4 129.5 109.2 ...  
## $ children\_weight: num [1:192] 19.6 13.9 10.5 23.6 16 ...  
## $ children\_age : num [1:192] 12 8 6.5 13 7 17 16 11 17 8 ...  
## $ children\_male : int [1:192] 1 0 0 1 0 1 0 1 0 1 ...

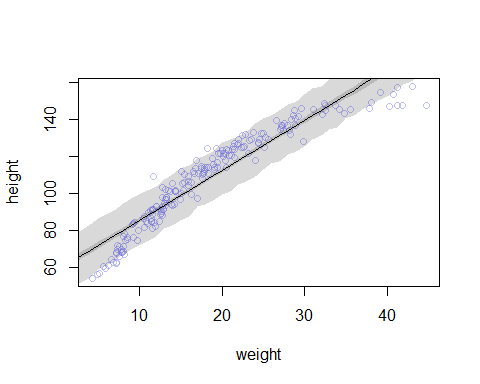
#Compute mean weight of data in d2  
children\_mean\_weight <- mean( d2$weight )  
children\_mean\_weight

## [1] 18.41419

#The mean weight of all of the children is 18.41419 kg.  
  
#Apply stan model  
model2 <- ulam(  
 alist(  
 children\_height ~ dnorm( mu , sigma ) ,  
 mu <- a + b\*( children\_weight - 18.41419 ) ,  
 a ~ dnorm( 178 , 20 ) ,  
 b ~ dlnorm( 0 , 1 ) ,  
 sigma ~ dunif( 0 , 50 )  
 ) ,  
 data=data\_list , chains=4 , cores=4 , iter=1000 )  
  
#Inspect the model  
precis( model2 )

## mean sd 5.5% 94.5% n\_eff Rhat  
## a 108.400569 0.64023872 107.358308 109.424976 2019.814 0.9993297  
## b 2.715830 0.06912315 2.604218 2.825069 1682.041 0.9999280  
## sigma 8.556862 0.45356047 7.876720 9.316452 1909.949 0.9996634

#Since the mean slope of the regression model is 2.72, we could expect that for every 10 units of increase in weight, the model would predict the child to be 27.2 units taller.  
  
#2-b  
#Create a sequence containing weights  
children.weight.seq <- seq( from=0 , to=50 , by=1 )  
  
#Create a list to be applied to Link function   
children.to.predict <- list( children\_weight=children.weight.seq )  
  
#Apply Link function to calculate mu  
children.mu.predict <- link( model2 , data=children.to.predict )  
  
#Calculate the mean and 89% interval of mu  
children.mu.predict.mean <- apply( children.mu.predict , 2 , mean )  
children.mu.predict.PI <- apply( children.mu.predict , 2 , PI , prob=0.89 )  
  
#Simulate predicted heights regarding each weight in children.to.predict  
children.height.predict <- sim( model2 , data=children.to.predict )  
  
#Calculate the 89% interval of predicted heights  
children.height.predict.PI <- apply( children.height.predict , 2 , PI , prob=0.89 )  
  
#Plot the results of model with raw data  
plot( height ~ weight , d2 , col=col.alpha(rangi2,0.5) )  
lines( children.weight.seq , children.mu.predict.mean )  
shade( children.mu.predict.PI , children.weight.seq )  
shade( children.height.predict.PI , children.weight.seq )



#2-c  
#From the graph above, we could find that although the 89% interval of predicted heights contained most of the raw data, the linear regression model itself did not fit the data well. It is possibly better to use polynomial regression model in this case, since the raw data seemed to be like a curve.

# Question3

#3-a  
#Create a data list for the model  
data\_list\_3 <- list(  
 data\_height = Howell1.data$height,  
 data\_weight = Howell1.data$weight,  
 data\_age = Howell1.data$age,  
 data\_male = Howell1.data$male  
)  
  
#Inspect the data list  
str(data\_list\_3)

## List of 4  
## $ data\_height: num [1:544] 152 140 137 157 145 ...  
## $ data\_weight: num [1:544] 47.8 36.5 31.9 53 41.3 ...  
## $ data\_age : num [1:544] 63 63 65 41 51 35 32 27 19 54 ...  
## $ data\_male : int [1:544] 1 0 0 1 0 1 0 1 0 1 ...

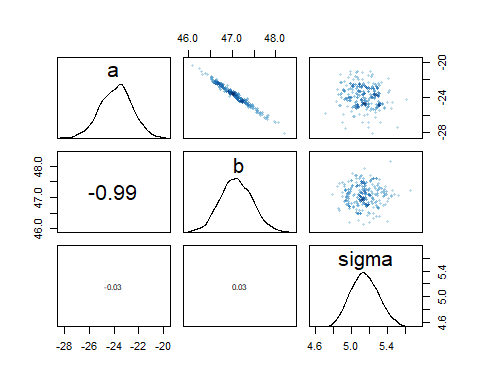
#Apply stan model  
model3 <- ulam(  
 alist(  
 data\_height ~ dnorm( mu , sigma ) ,  
 mu <- a + b\*log( data\_weight ) ,  
 a ~ dnorm( 178 , 100 ) ,  
 b ~ dnorm( 0 , 100 ) ,  
 sigma ~ dunif( 0 , 50 )  
 ) ,  
 data=data\_list\_3 , chains=4 , cores=4 , iter=1000 )  
  
#Inspect results of the model  
precis( model3 )

## mean sd 5.5% 94.5% n\_eff Rhat  
## a -23.836775 1.313120 -25.955821 -21.768387 603.8343 1.005726  
## b 47.092056 0.376396 46.510020 47.701410 564.7229 1.006261  
## sigma 5.150056 0.155069 4.908142 5.403849 1006.0175 1.001154

round( vcov( model3 ) , 3 )

## a b sigma  
## a 1.724 -0.487 -0.007  
## b -0.487 0.142 0.002  
## sigma -0.007 0.002 0.024

pairs( model3 )



#Interpret:  
#Since we are using a logarithm regression model to predict mu, it is quite diffucult to interpret the results. Because the logarithm is a non-linear regression model, the mean and standard deviation of a and b did not tell us a lot about the relationship between weights and heights. It would be better to do some sampling to know more about the model.   
  
#3-b  
#Beginning plot  
plot( height ~ weight , data=Howell1 , col=col.alpha(rangi2, 0.4) )  
  
#Create a sequence of weights and a list  
weight.seq3 <- seq( from=0 , to=70 , by=1 )  
weight.list <- list( data\_weight=weight.seq3 )  
  
#Apply Link function to generate values of mu  
predict.mu <- link( model3 , data=weight.list )  
  
#Calculate the mean and 97% interval of mu  
predict.mu.mean <- apply( predict.mu , 2 , mean )  
predict.mu.PI <- apply ( predict.mu , 2 , HPDI , prob=0.97 )  
  
#Apply Sim function to predict heights from values in weight.list  
predict.height <- sim( model3 , data=weight.list )  
  
#Calculate the 97% interval of predicted heights  
predict.height.PI <- apply( predict.height , 2 , HPDI , prob=0.97 )  
  
#Add the lines and shades to the plot  
lines( weight.seq3 , predict.mu.mean )  
shade( predict.mu.PI , weight.seq3 )  
shade( predict.height.PI , weight.seq3 )

