Computational Statistics & Probability

Lab 2 - Linear Models

Fall 2022

```
#knitr::opts_chunk$set(echo = TRUE)
#answer_key <- FALSE
library(rethinking)
```

```
## Loading required package: rstan
```

```
## Loading required package: StanHeaders
```

```
##
## rstan version 2.26.13 (Stan version 2.26.1)
```

```
## 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)
## For within-chain threading using `reduce_sum()` or `map_rect()` Stan functions,
## change `threads_per_chain` option:
## rstan_options(threads_per_chain = 1)
```

```
## Loading required package: cmdstanr
```

```
## This is cmdstanr version 0.5.3
```

```
## - CmdStanR documentation and vignettes: mc-stan.org/cmdstanr
```

```
## - CmdStan path: /Users/neelesh/.cmdstan/cmdstan-2.30.1
```

```
## - CmdStan version: 2.30.1
```

```
##
## A newer version of CmdStan is available. See ?install_cmdstan() to install it.
## To disable this check set option or environment variable CMDSTANR_NO_VER_CHECK=TRUE.
```

```
## Loading required package: parallel
```

```
## rethinking (Version 2.21)
```

```
##
## Attaching package: 'rethinking'
```

```
## The following object is masked from 'package:rstan':
##
##     stan
```

```
## The following object is masked from 'package:stats':
##
##     rstudent
```

```
library(DiagrammeR)
```

1. Predicting height from weight

Consider the following model from Chapter 4 of the textbook to use the !Kung census to predict height from weight of adults.

```
library(rethinking)
data(Howell1)
d <- Howell1
d2 <- d[ d$age >= 18 , ]
xbar <- mean(d2$weight)
m4.3 <- quap(
  alist(
    height ~ dnorm( mu , sigma ) ,
    mu <- a + b*( weight - xbar ),
    a ~ dnorm( 178 , 20 ),
    b ~ dlnorm( 0 , 1 ),
    sigma ~ dunif( 0 , 50 )
  ), data=d2 )
```

a) Using model `m4.3`, provide the predicted heights and 89% credibility intervals for each of the following individuals:

individual	weight	expected height	89% interval
1	47		
2	60		
3	37		
4	51		
5	43		

```
# First, create a dataframe of weights called `dat`
dat <- data.frame(individual = 1:5,  weight= c(47, 60, 37, 51, 43))
dat
```

```
##   individual weight
## 1          1     47
## 2          2     60
## 3          3     37
## 4          4     51
## 5          5     43
```

```
# Next, use the rethinking package `sim` function to run data through your
# model:
ht_sim <- sim( m4.3 , data = dat)
head(ht_sim)
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]
## [1,] 155.2787 168.1569 160.0060 154.5240 157.5223
## [2,] 159.6177 170.5661 139.6602 161.1455 158.3284
## [3,] 155.3663 171.6410 150.9070 165.3342 152.1219
## [4,] 159.2734 167.0829 145.8280 156.3095 149.4610
## [5,] 155.5862 164.2014 146.1204 156.4343 148.0871
## [6,] 152.9635 160.3828 141.5924 162.8761 154.3438
```

```
# Then calculate the expected heights from your simulation using apply, where
# `2` indicates the columns of ht_sim:
```

```r
Exp_ht <- apply(ht_sim, 2, mean)
Exp_ht
```

```
## [1] 156.4136 168.1617 147.5236 160.0982 152.7911
```

```r
# and the 89% credibility interval:
ht_CI <- apply(ht_sim, 2, PI, prob=0.89)
ht_CI
```

```
##         [,1]     [,2]     [,3]     [,4]     [,5]
## 5%  147.9700 160.2138 139.7114 152.2736 145.0091
## 94% 164.6365 176.2115 155.7739 168.4846 160.5437
```

```r
# These values can be added to the dataframe `dat` and displayed for an answer:
dat$Exp_ht <- Exp_ht
dat$CI89_L <- ht_CI[1,]
dat$CI89_U <- ht_CI[2,]
# The next table displays `round(dat, 1)`, which answers our questions:
```

```r
library(knitr)
kable(round(dat, 1 ), caption= "Predicted heights from weights")
```

Table 2: Predicted heights from weights

individual	weight	Exp_ht	CI89_L	CI89_U
1	47	156.4	148.0	164.6
2	60	168.2	160.2	176.2
3	37	147.5	139.7	155.8
4	51	160.1	152.3	168.5
5	43	152.8	145.0	160.5

b) Plot your regression line and 89% credibility interval.

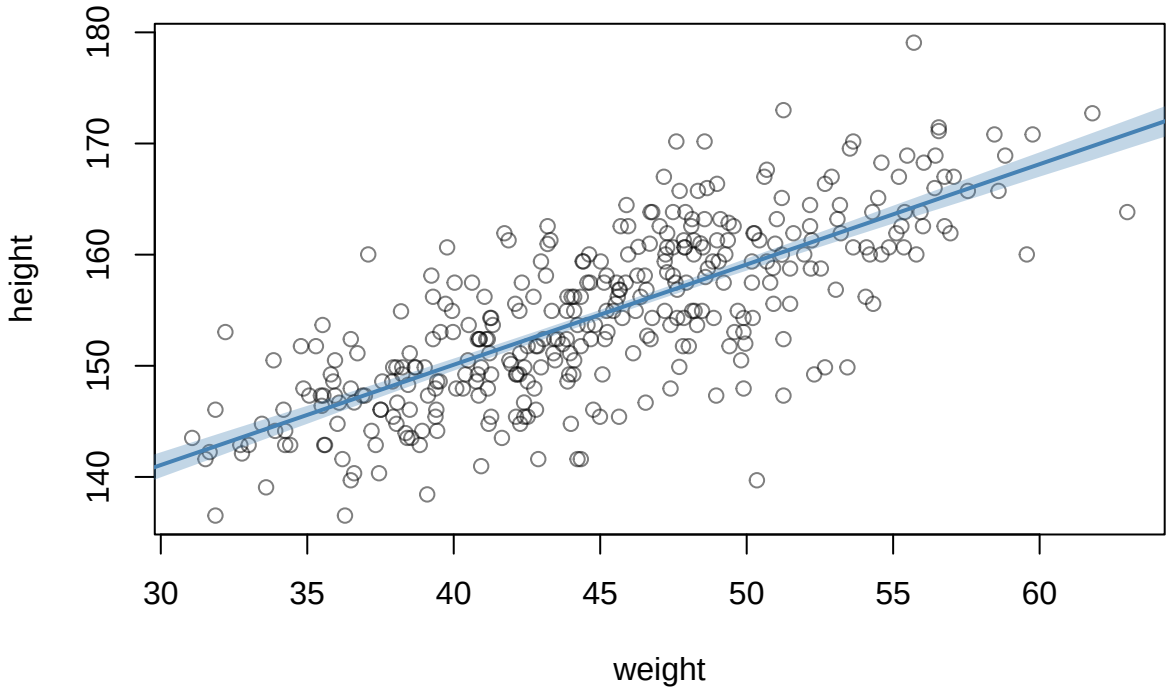
```r
# define a sequence of weights
weight.seq <- seq( from=25, to=70, by=1 )

# use link to compute mu for each sample from the posterior
# and for each weight in `weight.seq`
mu <- link( m4.3 , data=data.frame(weight=weight.seq) )

# summarize the distribution of mu
mu.mean <-apply( mu , 2, mean )  # mean of each column (axis '2') of the
                                 # matrix mu

# compute the 89% PI credibility interval
mu.PI <- apply( mu, 2, PI, prob=0.89 )

# plot data
plot( height ~ weight , data=d2, col=col.alpha("black",0.5) )
#plot MAP line, aka the mean mu for each weight
lines( weight.seq, mu.mean, col="steelblue", lwd=2)
#plot a shaded region for 89% PI
shade( mu.PI , weight.seq, col=col.alpha("steelblue",0.33) )
```


height
140
150
160
170
180
30
35
40
45
50
55
60
weight