# hw7 for stat341

*Zhihong Zhang*

*Feb 24th,2017*

Q: 5E2-3 5M4 5H1-2

5E2. Write down a multiple regression to evaluate the claim: Animal diversity is linearly related to latitude, but only after controlling for plant diversity. You just need to write down the model definition.

*Solution:*

using the fake sigma since missing data
$Animal\ diversity \sim Normal(\mu, \sigma)$
$\mu \sim a + b.latitude * latitude + c.pdiv * pdiv$
$\sigma \sim Uniform(0, 10)$

5E3. Write down a multiple regression to evaluate the claim: Neither amount of funding nor size of laboratory is by itself a good predictor of time to PhD degree; but together these variables are both positively associated with time to degree. Write down the model definition and indicate which side of zero each slope parameter should be on.

*Solution:*

$time\ to\ PhD\ degree \sim Normal(\mu, \sigma)$
$\mu \sim a + b * fundingsize * laboratory$
$\sigma \sim Uniform(0, 10)$

where b is a negative value for the slope. Since with larger funding and labortory, it took less time to get PHD degree.

5M4. In the divorce data, States with high numbers of Mormons (members of The Church of Jesus Christ of Latter-day Saints, LDS) have much lower divorce rates than the regression models expected. Find a list of LDS population by State and use those numbers as a predictor variable, predicting divorce rate using marriage rate, median age at marriage, and percent LDS population (possibly standardized). You may want to consider transformations of the raw percent LDS variable.

*Solution:*

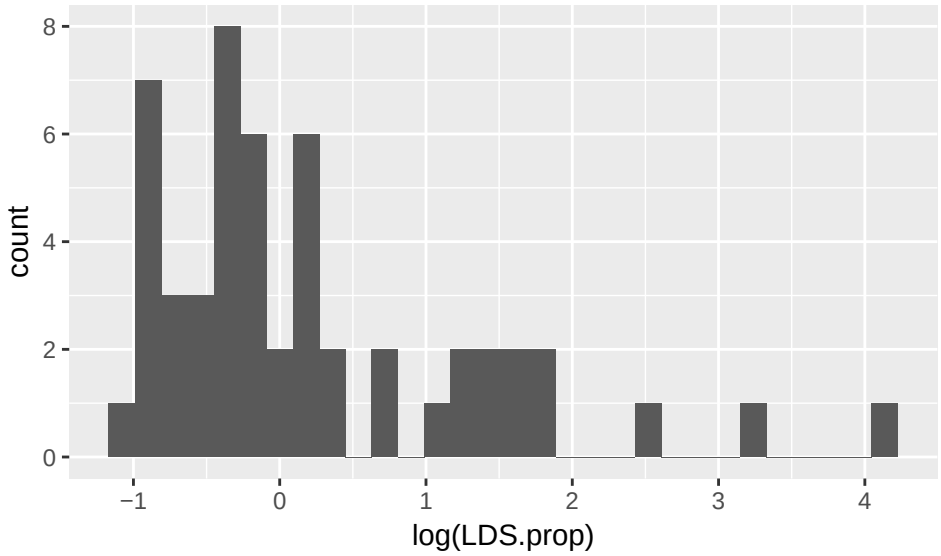
```
# doing this a crazy way to make it fit on the page
url <-
  paste0("https://en.wikipedia.org/wiki/",
         "The_Church_of_Jesus_Christ_of_Latter-day_Saints_",
         "membership_statistics_(United_States)"
  )
tables <- html_nodes(read_html(url), "table")
MormonsRaw <- html_table(tables[2], fill = TRUE)[[1]]


Mormons <-
  MormonsRaw %>%
  rename(LDS.prop = LDS) %>%
  mutate(
    Membership = parse_number(Membership),
    Population = parse_number(Population) / 1e6, # in millions
    LDS.prop = parse_number(LDS.prop)
```

```
) %>%
select(1:4) # keep only first four columns
```

```
require(statisticalModeling)
gf_histogram( ~ log(LDS.prop), data = Mormons)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




Based on the model, the Utah and Idaha has much more higher Mormon population rate than other states. Compared with these two data, the other state ratio is almost zero. Therefore usinglogrithm to decrese the difference between these states based on Mormon population effect. Therefore it is not neccseary to do transformation.

5H1. Fit two bivariate Gaussian regressions, using map: (1) body weight as a linear function of territory size (area), and (2) body weight as a linear function of groupsize. Plot the results of these regressions, displaying the MAP regression line and the 95% interval of the mean. Is either variable important for predicting fox body weight?

*Solution:*

```
data("foxes")

linearmodel1 <- map(alist(
weight ~ dnorm( mu , sigma ),
mu <- a + b*area,
a ~ dnorm( 10, 1 ),
b ~ dnorm( 0 , 4 ),
sigma ~ dunif( 0, 10 )
), data =foxes )


linearmodel1.pred <-
  data_frame(
    area = seq(from = 1, to = 7, by = .4)
  )

mu <- link(linearmodel1, data = linearmodel1.pred)
```

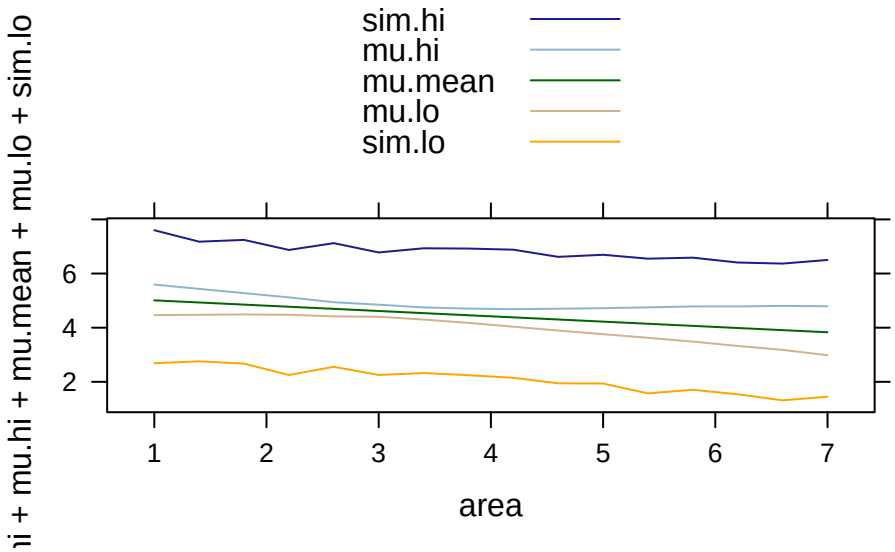
```
## [ 100 / 1000 ]
```

```
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

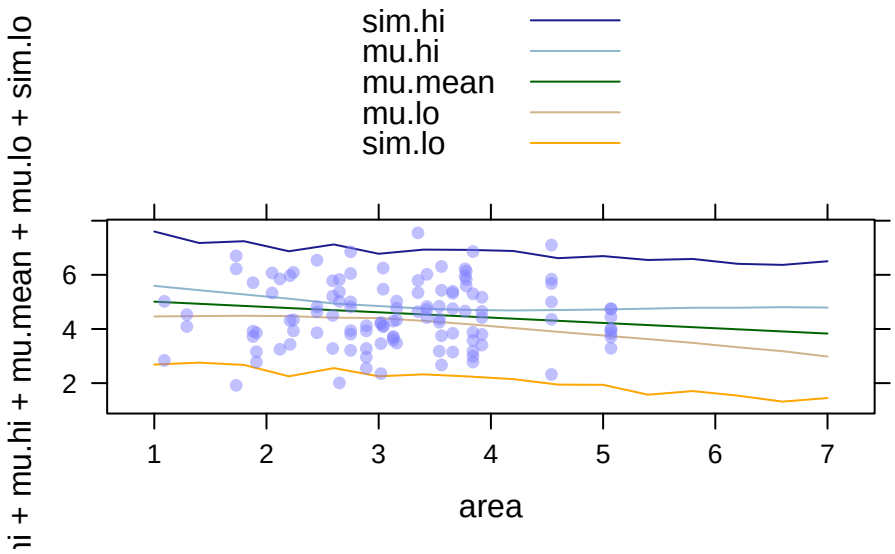
```
sim.area <- sim(linearmodel1, data = linearmodel1.pred)
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
linearmodel1.pred <-
  linearmodel1.pred %>%
  mutate(
    mu.mean = apply(mu, 2, mean),
    mu.lo = apply(mu, 2, HPDI,prob=0.95)[1,],
    mu.hi = apply(mu, 2, HPDI,prob=0.95)[2,],
    sim.lo = apply(sim.area, 2, HPDI,prob=0.95)[1,],
    sim.hi = apply(sim.area, 2, HPDI,prob=0.95)[2,]
    )
xyplot(sim.hi + mu.hi + mu.mean + mu.lo + sim.lo ~ area,
       data = linearmodel1.pred,  type = "l", auto.key = list(lines = TRUE, points = FALSE))
```




```
plotPoints(weight ~ area, data = foxes, col = rangi2, alpha = 0.5, add = TRUE)
```




```
#model 2


linearmodel2 <- map(alist(
weight ~ dnorm( mu , sigma ),
mu <- a + b*groupsize,
a ~ dnorm( 10, 1 ),
b ~ dnorm( 0 , 4 ),
sigma ~ dunif( 0, 10 )
), data =foxes )

linearmodel2.pred <-
  data_frame(
    groupsize = seq(from = 1, to = 10, by = 1)
  )

mu <- link(linearmodel2, data = linearmodel2.pred)
```

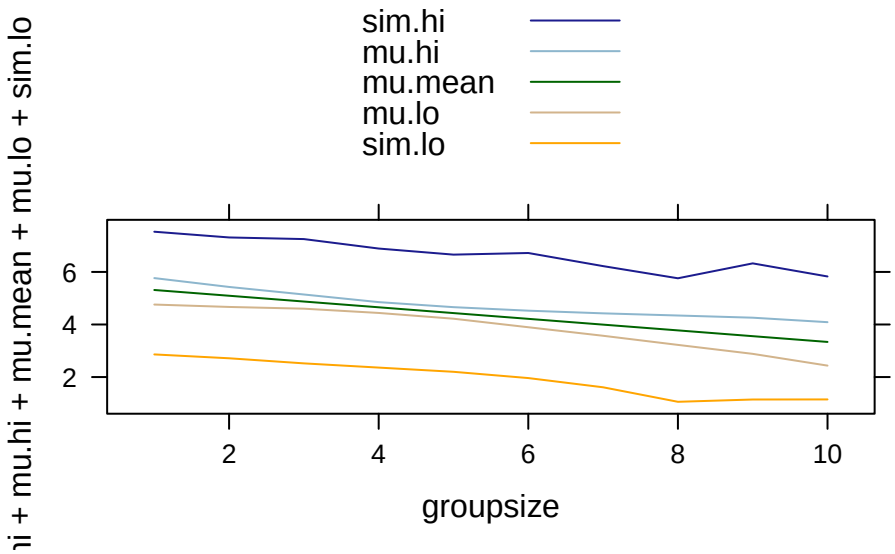
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
sim.groupsize <- sim(linearmodel2, data = linearmodel2.pred)
```

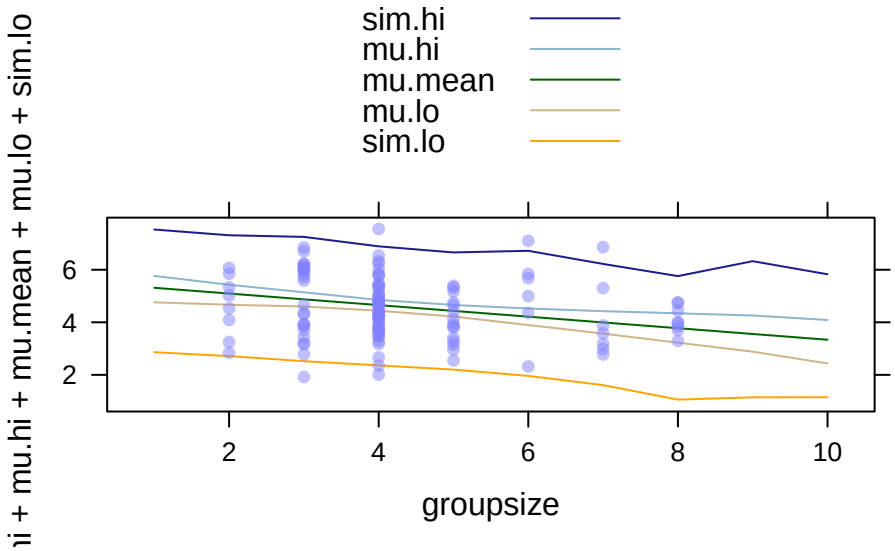
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
```

```
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```r
linearmodel2.pred <-
  linearmodel2.pred %>%
  mutate(
    mu.mean = apply(mu, 2, mean),
    mu.lo = apply(mu, 2, HPDI,prob=0.95)[1,],
    mu.hi = apply(mu, 2, HPDI,prob=0.95)[2,],
    sim.lo = apply(sim.groupsize, 2, HPDI,prob=0.95)[1,],
    sim.hi = apply(sim.groupsize, 2, HPDI,prob=0.95)[2,]
    )
xyplot(sim.hi + mu.hi + mu.mean + mu.lo + sim.lo ~ groupsize,
       data = linearmodel2.pred,  type = "l", auto.key = list(lines = TRUE, points = FALSE))
```




```r
plotPoints(weight ~ groupsize, data = foxes, col = rangi2, alpha = 0.5, add = TRUE)
```




both variables are important for predicting fox body weight.

5H2. Now fit a multiple linear regression with weight as the outcome and both area and groupsize as predictor variables. Plot the predictions of the model for each predictor, holding the other predictor constant at its mean. What does this model say about the importance of each variable? Why do you get different results than you got in the exercise just above?

*Solution:*

hold area constant

```
data("foxes")
linearmodel <- map(alist(
weight ~ dnorm( mu , sigma ),
mu <- a + b*area+c*groupsize,
a ~ dnorm( 10, 1 ),
b ~ dnorm( 0 , 4 ),
c ~ dnorm( 0 , 4 ),
sigma ~ dunif( 0, 10 )
), data =foxes )

linearmodel.pred <-
  data_frame(
  groupsize = seq(from = 1, to = 10, by = 1),
  area= mean(foxes$area)
  )

mu <- link(linearmodel, data = linearmodel.pred)
```

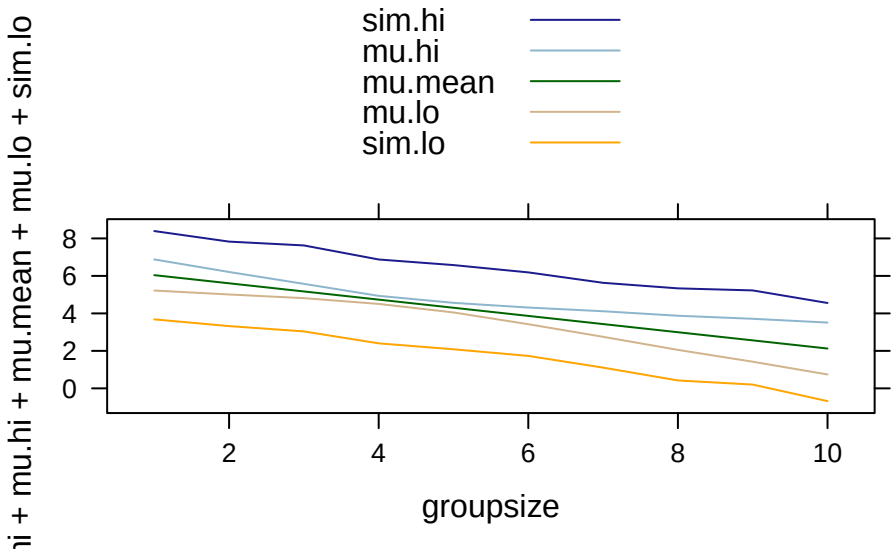
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
```

```
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

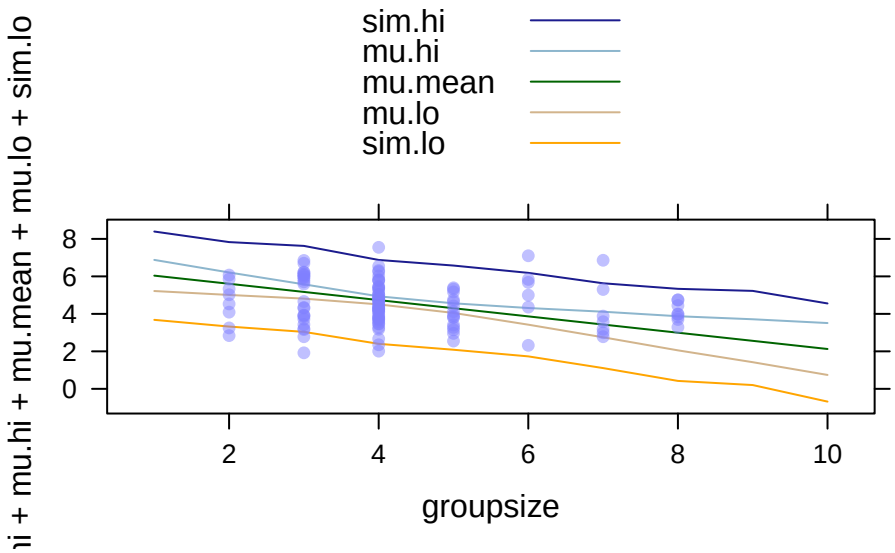
```
sim.groupsize <- sim(linearmodel, data = linearmodel.pred)
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
linearmodel.pred <-
  linearmodel.pred %>%
  mutate(
    mu.mean = apply(mu, 2, mean),
    mu.lo = apply(mu, 2, HPDI,prob=0.95)[1,],
    mu.hi = apply(mu, 2, HPDI,prob=0.95)[2,],
    sim.lo = apply(sim.groupsize, 2, HPDI,prob=0.95)[1,],
    sim.hi = apply(sim.groupsize, 2, HPDI,prob=0.95)[2,]
    )
xyplot(sim.hi + mu.hi + mu.mean + mu.lo + sim.lo ~ groupsize,
       data = linearmodel.pred,  type = "l", auto.key = list(lines = TRUE, points = FALSE))
```




```
plotPoints(weight ~ groupsize, data = foxes, col = rangi2, alpha = 0.5, add = TRUE)
```




hold size constant

```
data("foxes")
linearmodel <- map(alist(
weight ~ dnorm( mu , sigma ),
mu <- a + b*area+c*groupsize,
a ~ dnorm( 10, 1 ),
b ~ dnorm( 0 , 4 ),
c ~ dnorm( 0 , 4 ),
sigma ~ dunif( 0, 10 )
), data =foxes )

linearmodel.pred <-
  data_frame(
  area = seq(from = 1, to = 7, by = .5),
  groupsize= mean(foxes$groupsize)
  )

mu <- link(linearmodel, data = linearmodel.pred)
```

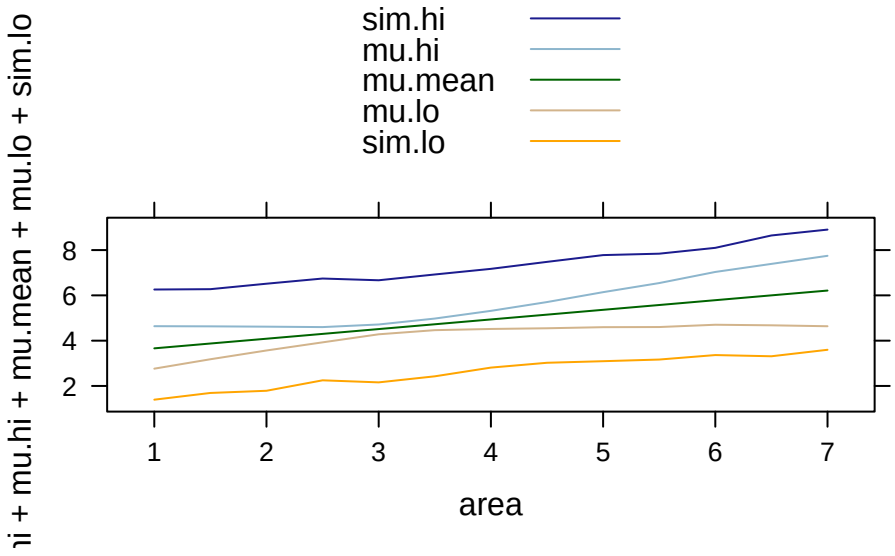
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
sim.area <- sim(linearmodel, data = linearmodel.pred)
```

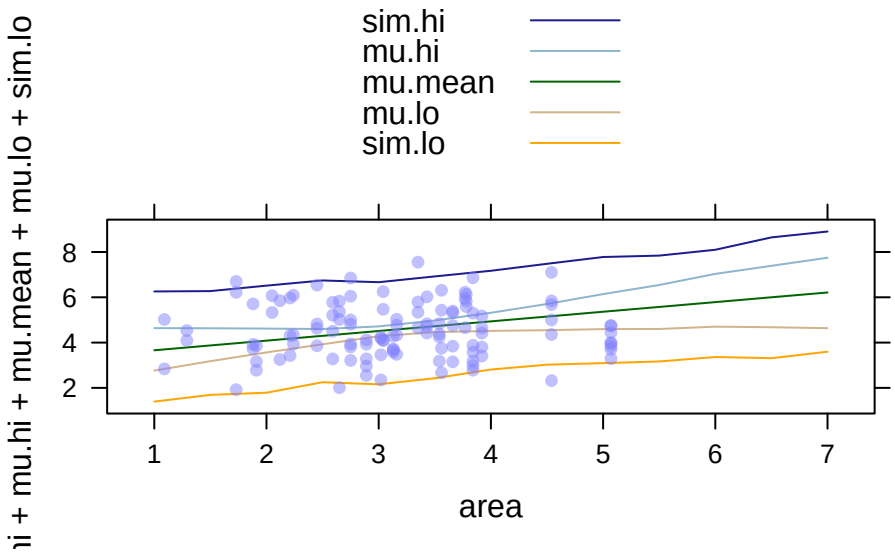
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
```

```
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```r
linearmodel.pred <-
  linearmodel.pred %>%
  mutate(
    mu.mean = apply(mu, 2, mean),
    mu.lo = apply(mu, 2, HPDI,prob=0.95)[1,],
    mu.hi = apply(mu, 2, HPDI,prob=0.95)[2,],
    sim.lo = apply(sim.area, 2, HPDI,prob=0.95)[1,],
    sim.hi = apply(sim.area, 2, HPDI,prob=0.95)[2,]
    )
xyplot(sim.hi + mu.hi + mu.mean + mu.lo + sim.lo ~ area,
       data = linearmodel.pred,  type = "l", auto.key = list(lines = TRUE, points = FALSE))
```




```r
plotPoints(weight ~ area, data = foxes, col = rangi2, alpha = 0.5, add = TRUE)
```




Based on the models, weight increase as the area increase while weight increase as the groupsize decrease.