STAT 447C: Bayesian Statistics **Fall 2024**

Exercise 4

Junsong Tang *junsong.tang@stat.ubc.ca*

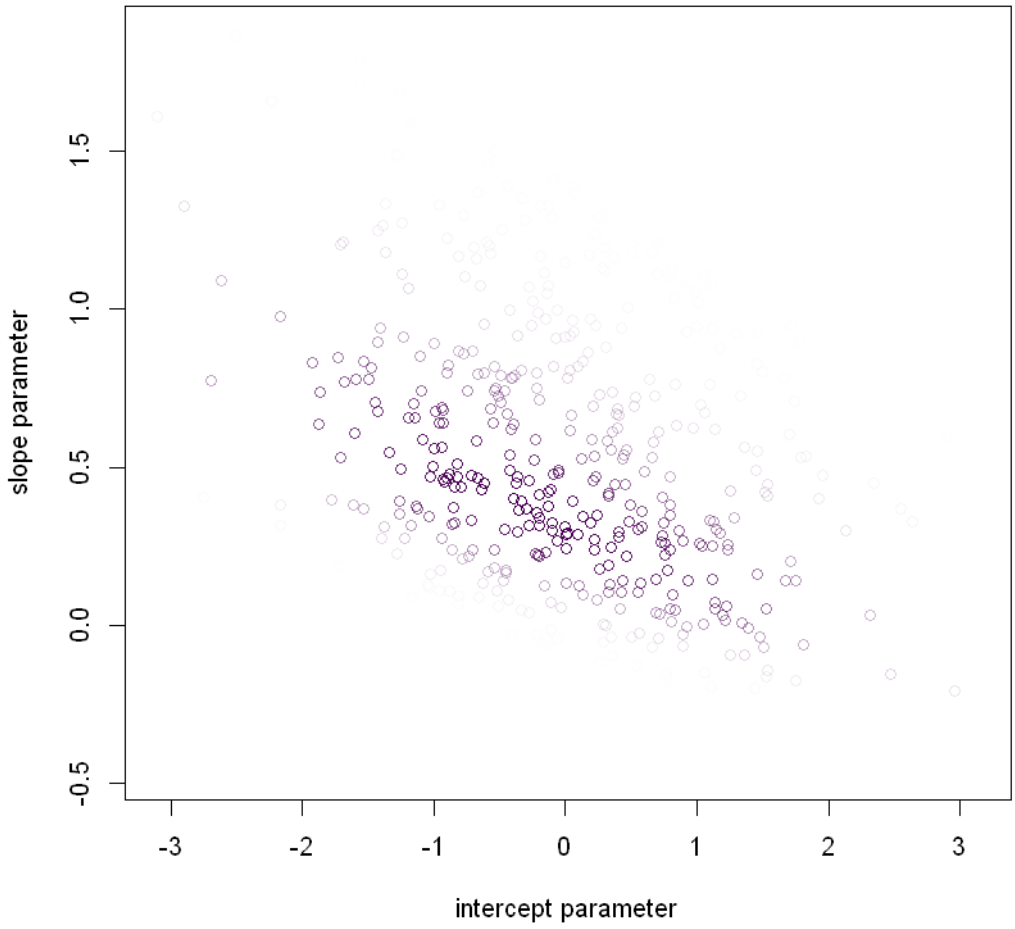
1 logistic rocket improvement

```r
suppressPackageStartupMessages(library(extraDistr))
suppressPackageStartupMessages(library(distr))
source("./simple.R")
source("./simple_utils.R")
set.seed(2024)

success_indicators = c(1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1)

# (1)
logistic_regression = function() {
  n = length(success_indicators)
  intercept = simulate(Norm(0, 1))
  slope = simulate(Norm(0, 1))
  thetas = plogis(intercept + slope * (1:n))
  for (i in (1:n)) {
    observe(success_indicators[i], Bern(thetas[i]))
  }
  theta_next = plogis(intercept + slope * (n + 1))
  y_next = simulate(Bern(theta_next))
  return (c(intercept, slope, y_next))
}
```

```r
# (2)
posterior = posterior_particles(logistic_regression, 1000)
weighted_scatter_plot(posterior, plot_options = list(xlab="intercept parameter", ylab="
    slope parameter"))
```



3. We wish to estimate $\mathbb{P}(Y_{n+1} = 1 | Y^{(n)} = y^{(n)})$, which is $\mathbb{E}(Y_{n+1} | Y^{(n)} = y^{(n)})$, so we can use the posterior function to do the calculation.

```
# (3)
post_obj = posterior(logistic_regression, 1000)
intercept = post_obj[1]
slope = post_obj[2]
pred = post_obj[3]
pred
# 0.951208300707776
```

4. Similar to the last question, we use posterior to compute the posterior mean of Y_{n+1} under the model with no slope.

```
# (4)
logistic_regression_2 = function() {
  n = length(success_indicators)
  intercept = simulate(Norm(0, 1))
  theta = plogis(intercept)
  for (i in (1:n)) {
    observe(success_indicators[i], Bern(theta))
  }
  y_next = simulate(Bern(theta))
  return (c(intercept, y_next))
}
(posterior(logistic_regression_2, 1000))[2]
```

```
# 0.730559041612286
```

2 choosing a model

This time, the model index is our parameter, and it has prior on Unif($\{0, 1\}$). Given the model index, we will use the corresponding model. we want to infer under the observation,the probability of the model index equals to 1, or equivalently, the posterior mean of model index, so that we could determine whether model 1 has higher chance or not. So we will implement a new ppl program and then use posterior function to compute the posterior mean.

```
unified_model = function() {
  # model ~ unif({0,1}) and it represents if we choose the model with slope(=1), or without
    slope(=0)
  model = simulate(Bern(1 / 2))
  if (model == 1) {
      logistic_regression()
  }
  else {
      logistic_regression_2()
  }
  return(model)
}
posterior(unified_model, 1000)
# 0.533340959655548
```

From above, we can see that under the observation, model index being 1 has higher probability, thus we should choose the model with slope.