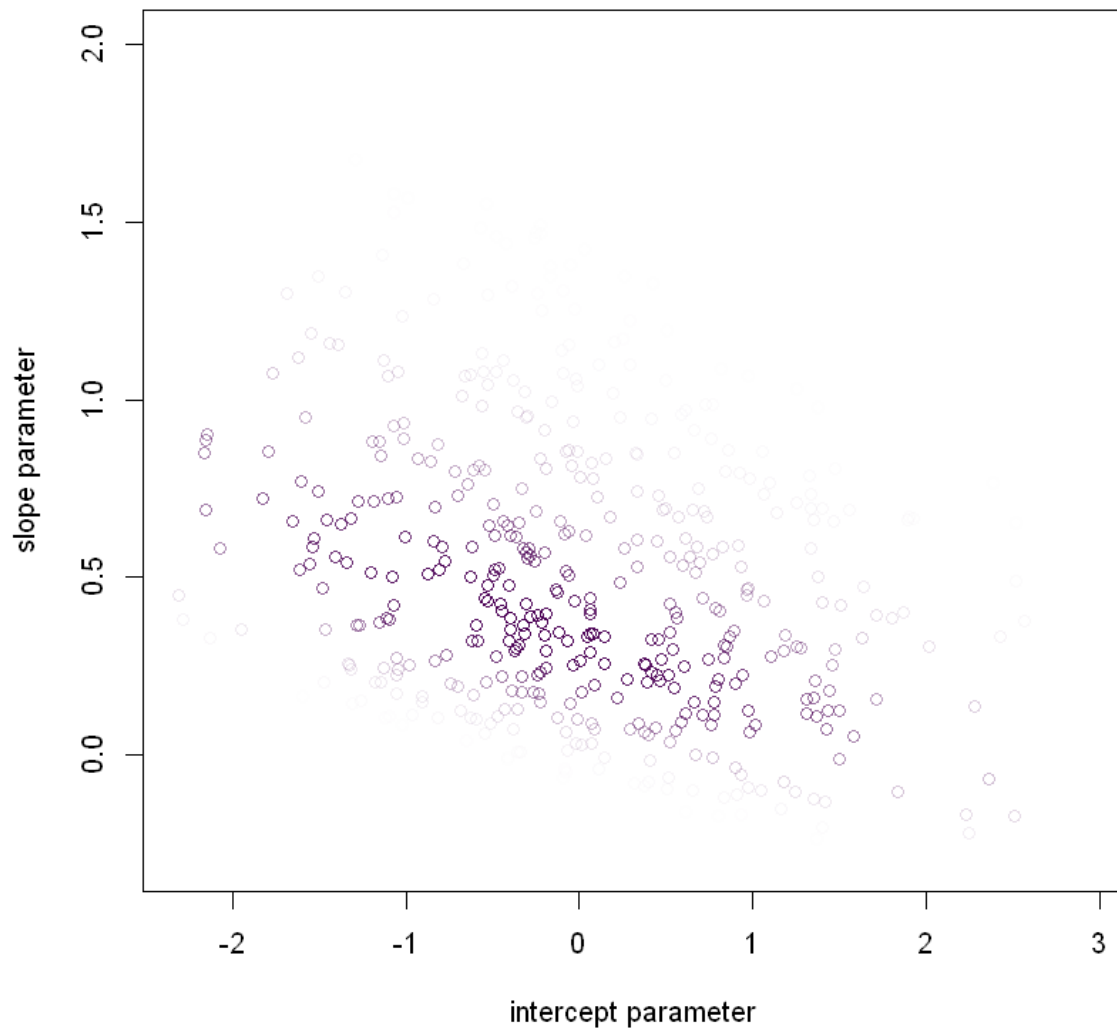


Exercise 4

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

1 logistic rocket improvement

```
1 suppressPackageStartupMessages(library(extraDistr))
2 suppressPackageStartupMessages(library(distr))
3 source("./simple.R")
4 source("./simple_utils.R")
5 set.seed(2024)
6
7 success_indicators = c(1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1)
8
9 # (1)
10 logistic_regression = function() {
11   n = length(success_indicators)
12   intercept = simulate(Norm(0, 1))
13   slope = simulate(Norm(0, 1))
14   thetas = plogis(intercept + slope * (1:n))
15   for (i in (1:n)) {
16     observe(success_indicators[i], Bern(thetas[i]))
17   }
18   next_theta = intercept + slope * (n + 1)
19   pred = ifelse(next_theta >= 1 / 2, 1, 0)
20   return (c(intercept, slope, pred))
21 }
22
23 # (2)
24 posterior = posterior_particles(logistic_regression, 1000)
25 weighted_scatter_plot(posterior, plot_options = list(xlab="intercept parameter", ylab="
  slope parameter"))
```



```

31 # (3)
32 post_obj = posterior(logistic_regression, 1000)
33 intercept = post_obj[1]
34 slope = post_obj[2]
35 plogis(slope * (length(success_indicators) + 1) + intercept)
36 # 0.990931632727317

```

```

41 # (4)
42 logistic_regression_2 = function() {
43   n = length(success_indicators)
44   intercept = simulate(Norm(0, 1))
45   theta = plogis(intercept)
46   for (i in (1:n)) {
47     observe(success_indicators[i], Bern(theta))
48   }
49   pred = ifelse(theta >= 1 / 2, 1, 0)
50   return (c(intercept, pred))
51 }
52 plogis((posterior(logistic_regression_2, 1000))[1])
53 # 0.733947433558989

```

2 choosing a model