# 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]))
  }
  next_theta = intercept + slope * (n + 1)
  pred = ifelse(next_theta >= 1 / 2, 1, 0)
  return (c(intercept, slope, pred))
}
```

```r
# (2)
posterior = posterior_particles(logistic_regression, 1000)
weighted_scatter_plot(posterior, plot_options = list(xlab="intercept parameter", ylab="slope parameter"))
```
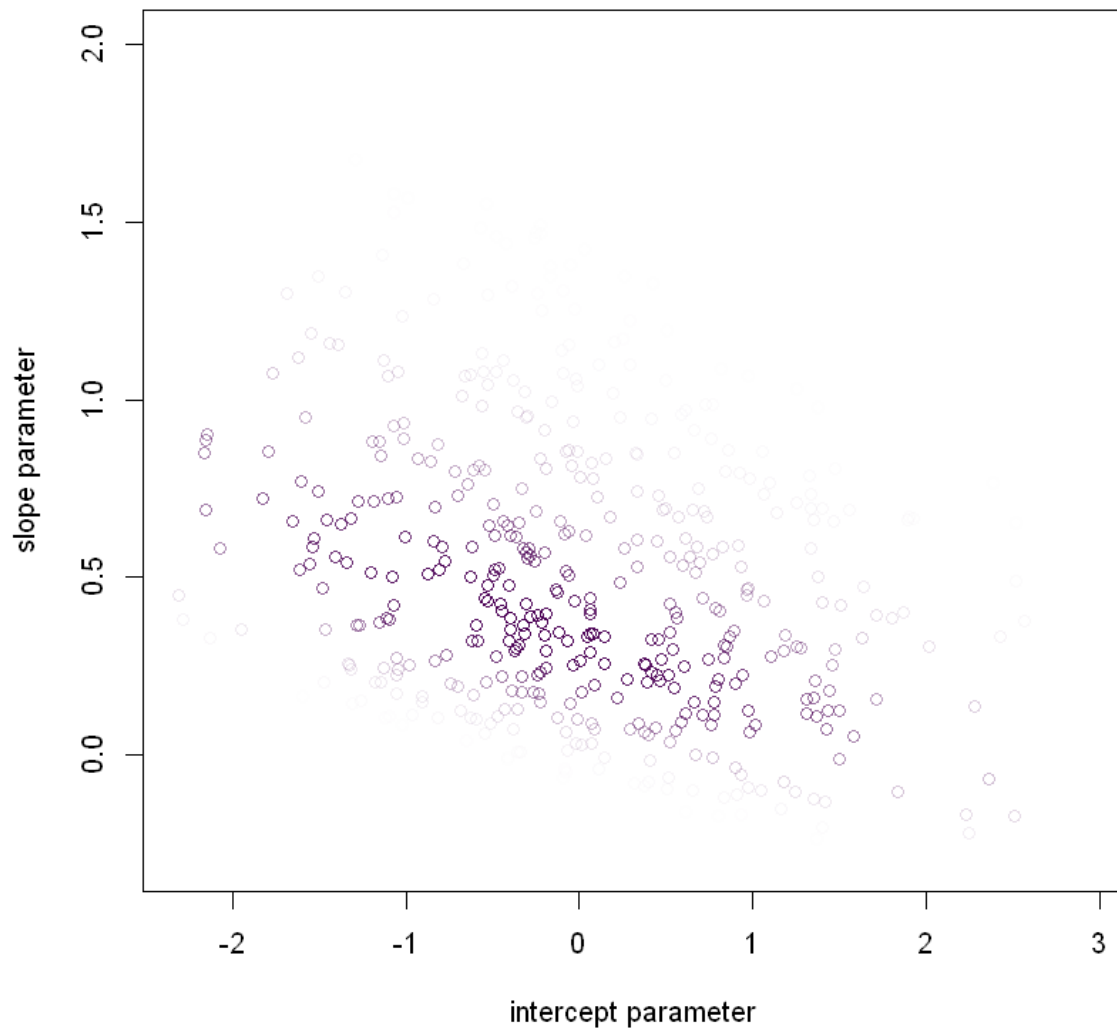
```
3  1  # (3)
   2  post_obj = posterior(logistic_regression, 1000)
   3  intercept = post_obj[1]
   4  slope = post_obj[2]
   5  plogis(slope * (length(success_indicators) + 1) + intercept)
   6  # 0.990931632727317
```

```
4  1  # (4)
   2  logistic_regression_2 = function() {
   3    n = length(success_indicators)
   4    intercept = simulate(Norm(0, 1))
   5    theta = plogis(intercept)
   6    for (i in (1:n)) {
   7      observe(success_indicators[i], Bern(theta))
   8    }
   9    pred = ifelse(theta >= 1 / 2, 1, 0)
  10    return (c(intercept, pred))
  11  }
  12  plogis((posterior(logistic_regression_2, 1000))[1])
  13  # 0.733947433558989
```

2   choosing a model

```
1 unified_model = function() {
2   post_logis = logistic_regression()
3   post_simple = logistic_regression_2()
4   p_logis = plogis(post_logis[1] + post_logis[2] * length(success_indicators))
5   p_simple = plogis(post_simple[1])
6   return (ifelse(p_logis > p_simple, 1, 0))
7 }
8 posterior(unified_model, 1000)
9 # 0.955874166557784
```