STAT-S676

HW3

Problems 1, 2

$$k|n, \lambda \sim TruncPoisson(\lambda, [1, n])$$
$$\pi_1, ..., \pi_k|k \sim Dirichlet(1, ..., 1)$$
$$z_1, ..., z_n|\overrightarrow{\pi} \overset{iid}{\sim} Multinom(1, \{1, ..., k\}, \vec{\pi})$$

But the labels of the partitions are arbitrary. For example, $(1, 1, 2, 2)$ is identical to $(2, 2, 1, 1)$. Then

$$\overrightarrow{z}|k \sim MultinomDirichlet$$

with p.m.f.

$$f(\overrightarrow{z}|k) = f(C|k) = k!\frac{\Gamma(k)}{\Gamma(n+k)}\prod_{j=1}^{k}\Gamma(m_j + 1)$$

where m_j is the number of objects in cluster j, kk is the number of clusters, and C is some clustering with k clusters.

If instead of setting the number of clusters to k, we let k be the upper bound for the number of clusters, then the probability a clustering with $\ell \leq k$ clusters is given by taking the sum from $\ell = 1$ to k of the probabilities of clusterings with ℓ clusters (times the number of possible such clusterings). But instead, we might try something like \$P(m_1, ..., m_k where at least one is 0) $= 1 - P($no $m_j = 0) = \sum_{\ell}^{k-1} P(m_1, ..., m_k$ where exactly ℓ are 0). We eventually get (curteously of Dr. Womack):

$$P(C) = \sum_{k=\ell}^{n}\frac{k(k-1)\cdots(k-\ell+1)}{M(n)}\frac{\Gamma(k)}{\Gamma(n+k)}\prod_{j=1}^{\ell}\Gamma(m_j+1)\frac{\lambda^k}{k!}$$

where $M(n) = \sum_{j=1}^{n}\frac{\lambda^j}{j!}$ and $D(\ell, n) = \sum_{k=\ell}^{n}\frac{\lambda^{k-\ell}\Gamma(k)\Gamma(n+\ell)}{\Gamma(k-\ell+1)\Gamma(n+k)}$ (from the truncated Poisson part).

This is implemented in R per Dr. Womack's code:

```
# literally grabbing Dr. Womack's functions (i.e. not mine)
# removed everything but the functions
source('~/dev/stats-hw/stat-s676/dpcode_new_final.R')

# the overwrite the relevant part
log_prior <- function(C,log_theta=0, d=.5, lambda=1, type="DP"){
  summary_C <- summary(reduce_C(as.factor(C)))
  theta <- exp(log_theta)
  k <- length(summary_C)
  if (type == "DP") {
    # https://en.wikipedia.org/wiki/Chinese_restaurant_process
    lgamma(exp(log_theta)) -
      lgamma(exp(log_theta) + length(C)) +
      sum(log_theta + lgamma(summary_C))
```

```
  } else if (type == "PY") {
    # https://en.wikipedia.org/wiki/Chinese_restaurant_process
    # our d is their alpha
    theta <- exp(log_theta)
    lgamma(theta) - lgamma(theta + length(C)) +
      k * log(d) + lgamma(theta / d + k) - lgamma(theta / d) +
      sum(summary_C - d) - lgamma(1 - d)
  } else {
    # problem 1
    log_lambda <- log(lambda)
    n <- length(C)
    log_M <- log_M_fun(n, log_lambda)
    log_D <- log_D_fun(seq(n), n, log_lambda)
    log_prior_multdir(C, log_M, log_D, log_lambda)
  }
}
```

Problem 3

```
##############################################################################
# MOST OF THIS CODE IS PRETTY MUCH COPY-PASTED FROM DR. WOMACK'S EXAMPLE     #
# I.E. NOT MINE (AND I MIGHT NOT KNOW 100% OF WHAT'S GOING ON)               #
##############################################################################

# packages, etc.
import::from(magrittr, `%>%`, `%<>%`)
import::from(parallel, mclapply, detectCores)
library(ggplot2)

# other setup
theme_set(theme_bw())
options(mc.cores = detectCores())

# data
iris_list <- iris %>%
  dplyr::select(-Species) %>%
  as.list()

# parameters defined as they were in example
N <- 100
a <- b <- 1
theta <- 1
q <- .5
K <- 2

# cluster separately for each measurement "y"
clusterings <- mclapply(iris_list, function(y) {
  n <- length(y)

  # start with random assignments
  C <- sample(seq(K), n, replace = TRUE)
```

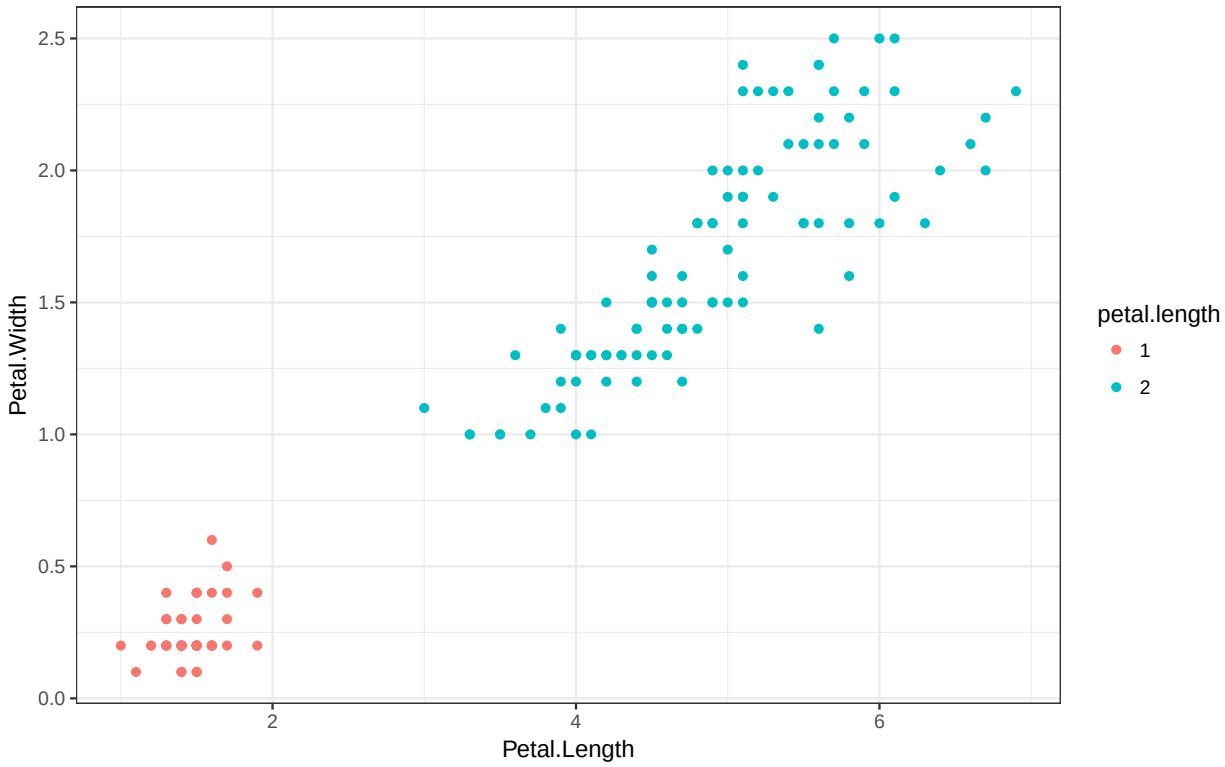
```
  C <- reduce_C(as.factor(C))

  C_out <- matrix(NA, N, n)
  for (j in seq(N)) {
    K <- length(levels(C))
    q <- sample_q(theta, n)
    theta <- sample_theta(K, q, a, b)
    log_theta <- log(theta)
    if (runif(1) < .5) {
      C <- sample_C_gibbs(C, y, log_theta)
    } else {
      C <- sample_C_split_merge(C, y, log_theta)
    }
    C_out[j, ] <- C
  }

  return(list(theta = theta, C = C, C_out = C_out))
})

# correspondence to species? probably not :(
lapply(clusterings, function(cl) {
  table(iris$Species, cl$C)
})
```

```
$Sepal.Length

            1
  setosa     50
  versicolor 50
  virginica  50

$Sepal.Width

            1
  setosa     50
  versicolor 50
  virginica  50

$Petal.Length

            1  2
  setosa     50  0
  versicolor  0 50
  virginica   0 50

$Petal.Width

            1  2
  setosa     49  1
  versicolor  0 50
  virginica   0 50
```

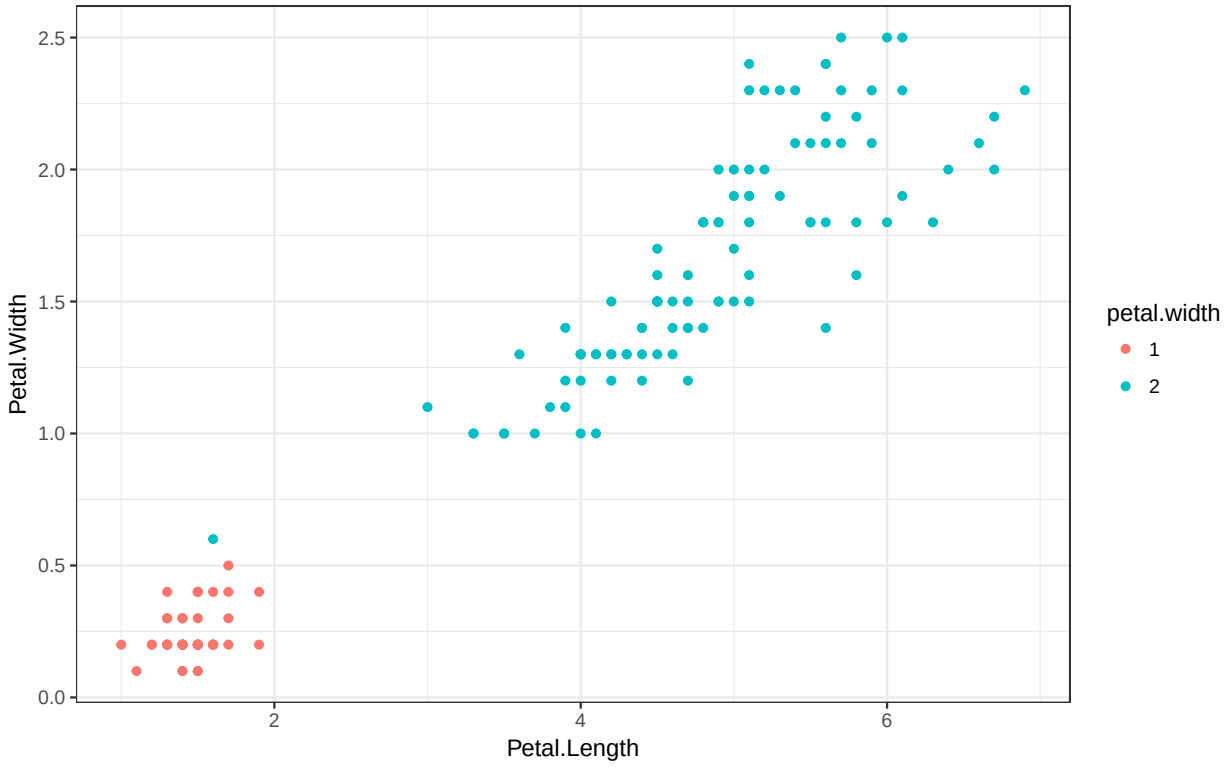
```
# thetas
sapply(clusterings, function(cl) cl$theta)
```

```
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
  0.20257199   0.11665336   0.08159633   0.34840590
```

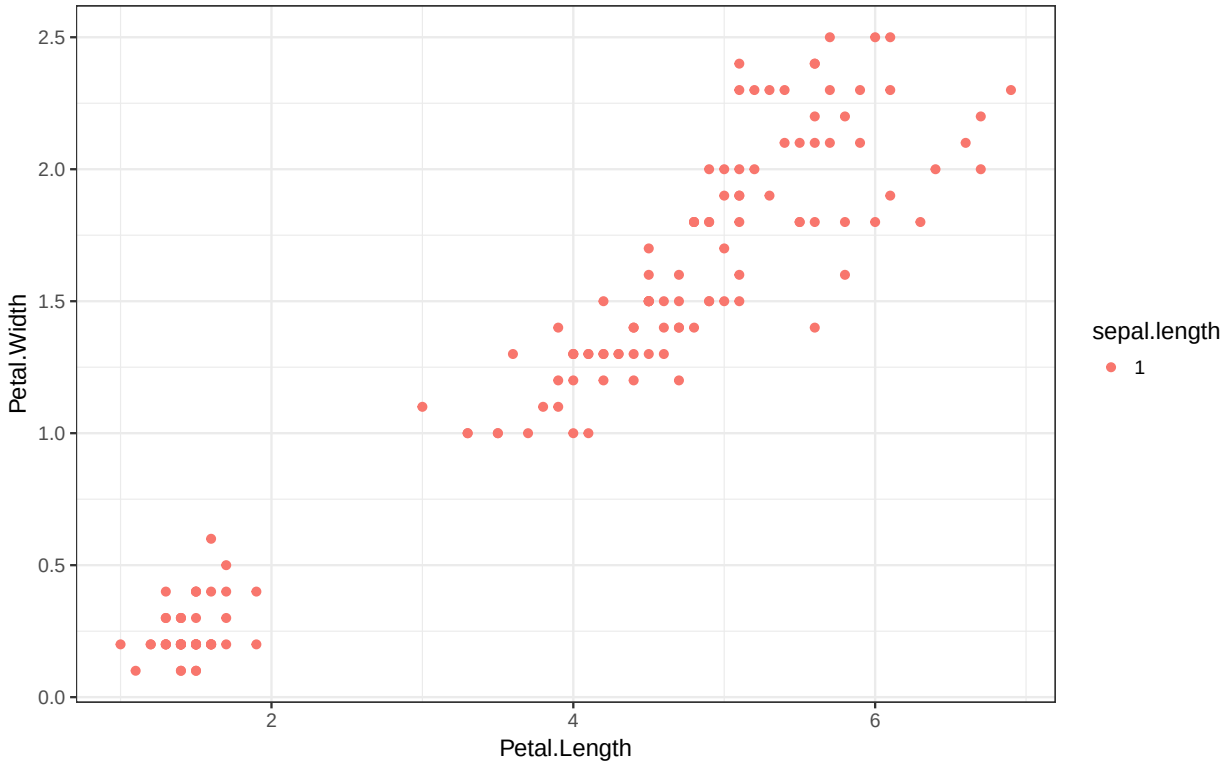
```
# viz
clusterings_df <- lapply(clusterings, function(cl) cl$C) %>%
  dplyr::bind_rows() %>%
  magrittr::set_colnames(tolower(colnames(.)))
iris %<>% dplyr::bind_cols(clusterings_df)
# i can do a loop for this oh well
ggplot(iris) +
  geom_point(aes(x = Petal.Length, y = Petal.Width, colour = petal.length))
```




```
ggplot(iris) +
  geom_point(aes(x = Petal.Length, y = Petal.Width, colour = petal.width))
```




```
ggplot(iris) +
  geom_point(aes(x = Petal.Length, y = Petal.Width, colour = sepal.length))
```



```
ggplot(iris) +
  geom_point(aes(x = Petal.Length, y = Petal.Width, colour = sepal.width))
```