STAT 574B
Homework 2

Nathan Hattersley

October 2, 2017

1 Poisson-Gamma Similarity

part a.

Let $\tilde{M}|\theta \sim \text{Poisson}(36\theta)$, $\theta \sim \text{Gamma}(\alpha, \beta)$. We know the posterior distribution of $\theta|Y$ from the examples we've done in class:

$$\theta|Y \sim \text{Gamma}(\alpha + \sum_i y_i, \beta + n)$$

We can then find the PPD of $\tilde{M}$ by taking $P\{\tilde{M}|\theta\}P\{\theta|y_1, \dots, y_n\}$ and integrating out theta:

$$\begin{aligned} P\{\tilde{M} = \tilde{m}|y\} &\propto \int_0^\infty (36\theta)^{\tilde{m}} e^{-36\theta} \theta^{\alpha + \sum_i y_i - 1} e^{-(\beta+n)\theta} d\theta \\ &\propto \int_0^\infty \theta^{\tilde{m} + \alpha + \sum_i y_i - 1} e^{-(\beta+n+36)\theta} d\theta \end{aligned}$$

With our sharp eyes and knowledge of conjugate analysis, we can identify the PPD of $\tilde{M}$ as a Negative Binomial with $n = \alpha + \sum_i y_i$, $p = \left(\frac{\beta+n}{\beta+n+36}\right)$.

part b.

We'll use Monte Carlo methods to sample from the posterior distribution and then plug into the distribution of $\tilde{M}|\theta$. I include an exact sample from the PPD to check my work, as it were.

```
set.seed(12)
a <- 0.5
b <- 0.5
n <- 12
sum_y <- 9

thetas <- rgamma(10000, a + sum_y, rate = b + n)
ppd.mc <- rpois(10000, 36 * thetas)

mean(ppd.mc)
```

```
## [1] 27.452
```

```
sd(ppd.mc)
```

```
## [1] 10.37103
```

```
ppd.exact <- rnbinom(10000, a + sum_y, (b + n)/(b + n + 36))
```

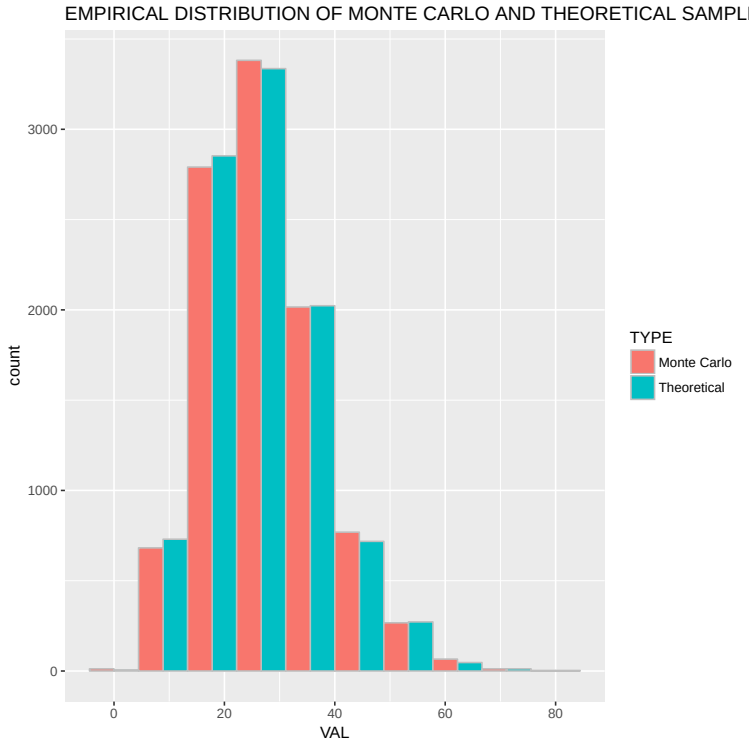
```
mean(ppd.exact)
```

```
## [1] 27.1627
```

```
sd(ppd.exact)
```

```
## [1] 10.29915
```

```
ggplot(data.frame(VAL = c(ppd.mc, ppd.exact), TYPE = rep(c("Monte Carlo",
    "Theoretical"), each = 10000))) + geom_histogram(aes(VAL,
    fill = TYPE), bins = 10, position = "dodge", col = "grey") +
    labs(title = "EMPIRICAL DISTRIBUTION OF MONTE CARLO AND THEORETICAL SAMPLES")
```



part c.

Here is a one-liner ready for the comedy festival:

```
(pr <- sum(ppd.mc >= 39)/10000)
```

```
## [1] 0.1431
```

There is approximately a 14.3% chance of getting enough patients in the next 36 months to complete the survey on time. I would hedge my bets were I a betting man.

2 Beta-Binomial Model

Assume X and Y are distributed with joint distribution:

$$p(x,y) \propto \binom{n}{x} y^{x+\alpha-1}(1-y)^{n-x+\beta-1}$$

With $X \in [0,1,\ldots,n],\ Y \in [0,1]$.

part a.

Let's do some integration/summation to get (proportionally) the marginals, then divide them out of the joint distribution to get conditional forms.

- $$\begin{aligned} p(x) &\propto \int_0^1 \binom{n}{k} y^{x+\alpha-1}(1-y)^{n-x+\beta-1} dy \\ &\propto \binom{n}{k} B(\alpha+x, n-x+\beta) \\ p(y|x) &\propto B(\alpha+x, n-x+\beta)^{-1} y^{x+\alpha-1}(1-y)^{n-x+\beta-1} \end{aligned}$$

- $$\begin{aligned} p(y) &\propto y^{\alpha-1}(1-y)^{\beta-1} \sum_{x=0}^{n} \binom{n}{k} y^x (1-y)^{n-x} \\ &\propto y^{\alpha-1}(1-y)^{\beta-1} \\ p(x|y) &\propto \binom{n}{x} y^x (1-y)^{n-x} \end{aligned}$$

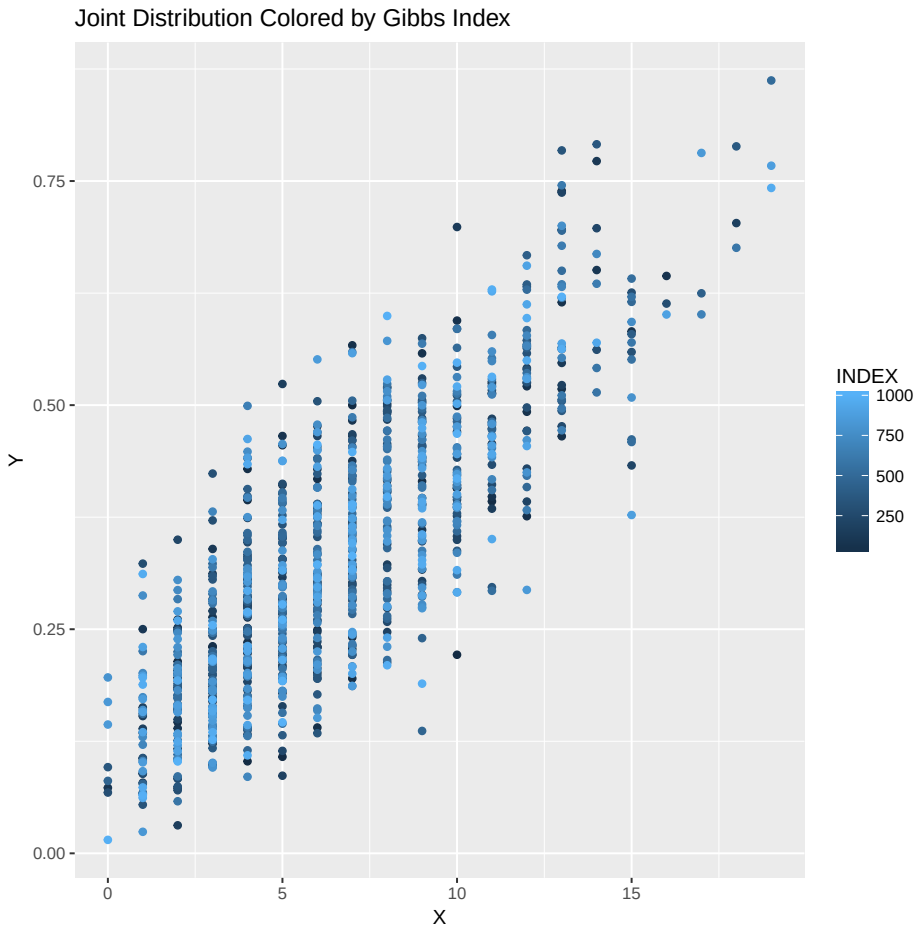
From this we can identify that $X|Y \sim \text{Binom}(n,x)$, and $Y|X \sim \text{Beta}(\alpha+y, \beta+n-x)$.

part b.

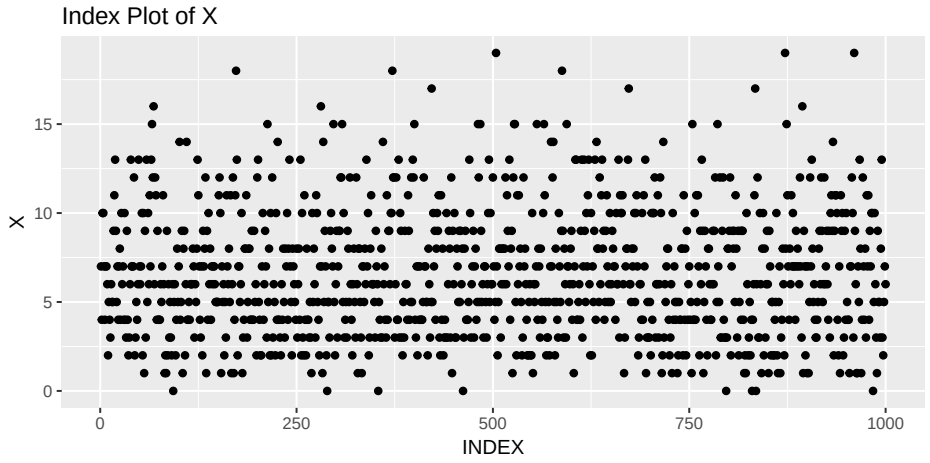
To do a Gibbs sample from these conditionals, I'll start by fixing α, β, and n. Then I'll pick a value of Y to start with, and take a varying number of samples. Then I'll make plots of X vs. Y, index plots, etc., then run acf and ESS to quantitatively assess the convergence of the sample.

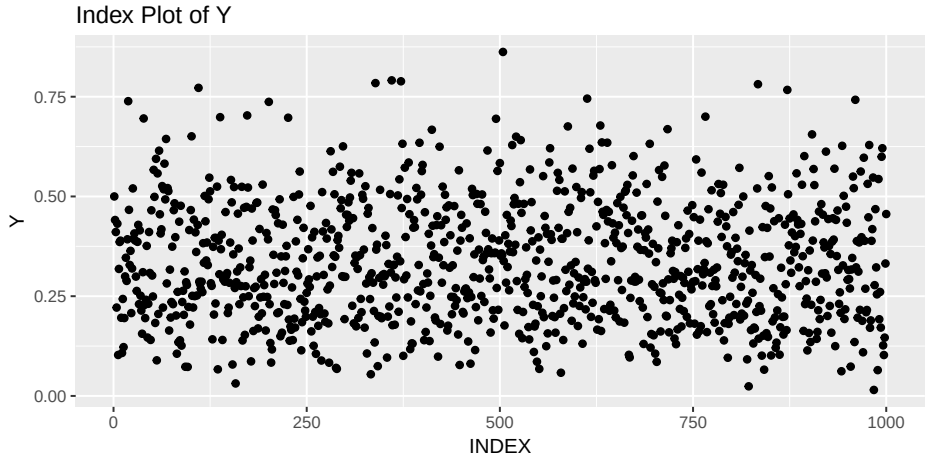
```
## Fixed parameters
set.seed(12)
a <- 3
b <- 6
n <- 20

## Fix Gibbs sample size
size <- 1e+05
## Initial value of Y|X
y_0 <- 0.5
## Thin factor
```

```r
thin <- 100


results.df <- data.frame(INDEX = 1:(size/thin), X = numeric(size/thin),
    Y = numeric(size/thin))

## Magrittr and within make for some sweet syntactic sugar
y <- y_0
x <- rbinom(1, n, y)
results.df %<>% within({
    Y[1] <- y
    X[1] <- x
})

## j is an index variable for results.df
j <- 2

for (i in 2:size) {

    y <<- rbeta(1, a + x, b + n - x)
    x <<- rbinom(1, n, y)

    if ((i - 1)%%thin == 0) {
        results.df %<>% within({
            Y[j] <- y
            X[j] <- x
        })
        j <<- j + 1
    }
}

summary(results.df[, c("Y", "X")])

##        Y                 X
##  Min.   :0.01502   Min.   : 0.00
##  1st Qu.:0.21954   1st Qu.: 4.00
##  Median :0.31647   Median : 6.00
##  Mean   :0.33533   Mean   : 6.61
##  3rd Qu.:0.43918   3rd Qu.: 9.00
##  Max.   :0.86222   Max.   :19.00

ggplot(results.df) + geom_point(aes(x = X, y = Y, color = INDEX)) +
    labs(title = "Joint Distribution Colored by Gibbs Index")
```

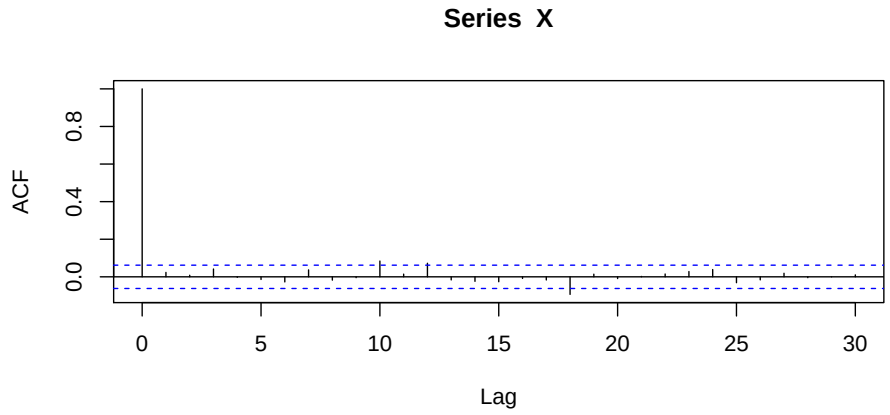


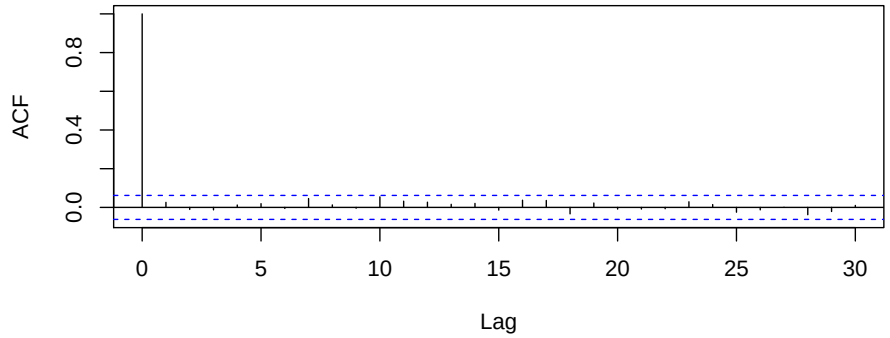

```
## INDEX PLOTS
(function() {
    p <- ggplot(results.df) + geom_point(aes(x = INDEX, y = X)) +
        labs(title = "Index Plot of X")
    q <- ggplot(results.df) + geom_point(aes(x = INDEX, y = Y)) +
        labs(title = "Index Plot of Y")
    grid.arrange(p, q)
})()
```



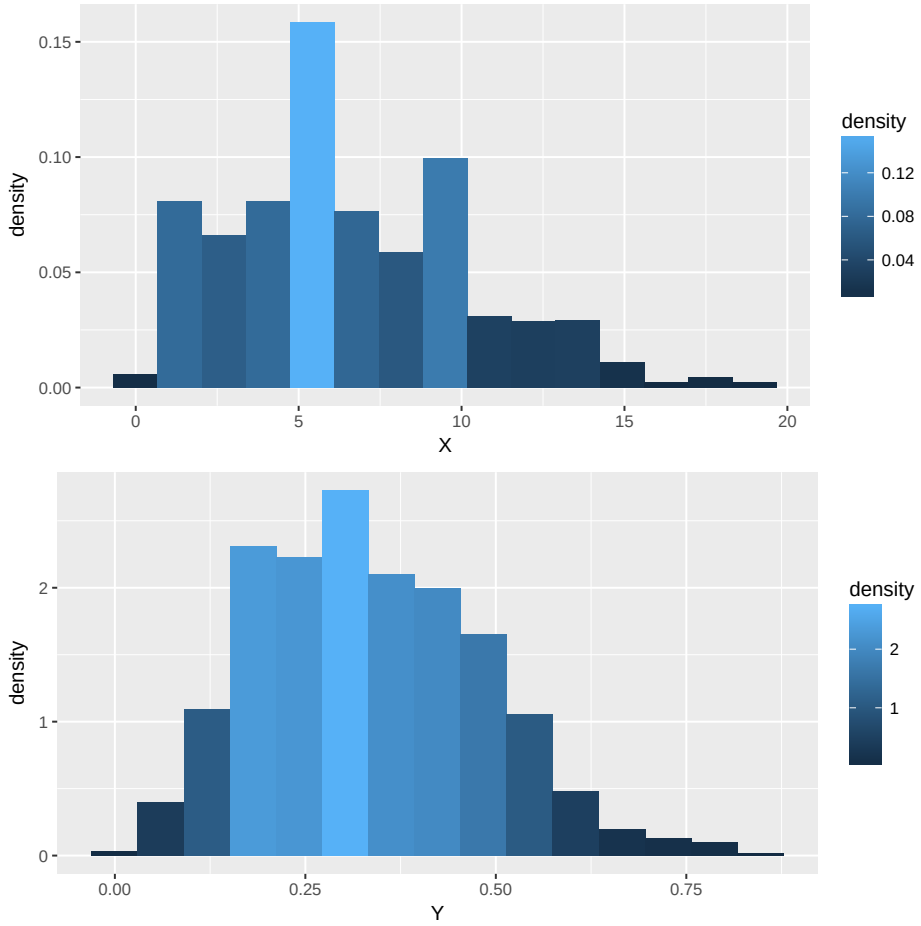



```
## ACF AND ESS
par(mfrow = c(2, 1))
with(results.df, {
    acf(X)
    acf(Y)
    c(effectiveSize(X), effectiveSize(Y))
})
```






```
## var1 var1
## 1000 1000

## HISTOGRAMS
(function() {
    p <- ggplot(results.df) + geom_histogram(aes(X, y = ..density..,
        fill = ..density..), bins = 15)
    q <- ggplot(results.df) + geom_histogram(aes(Y, y = ..density..,
        fill = ..density..), bins = 15)
    grid.arrange(p, q)
})()
```



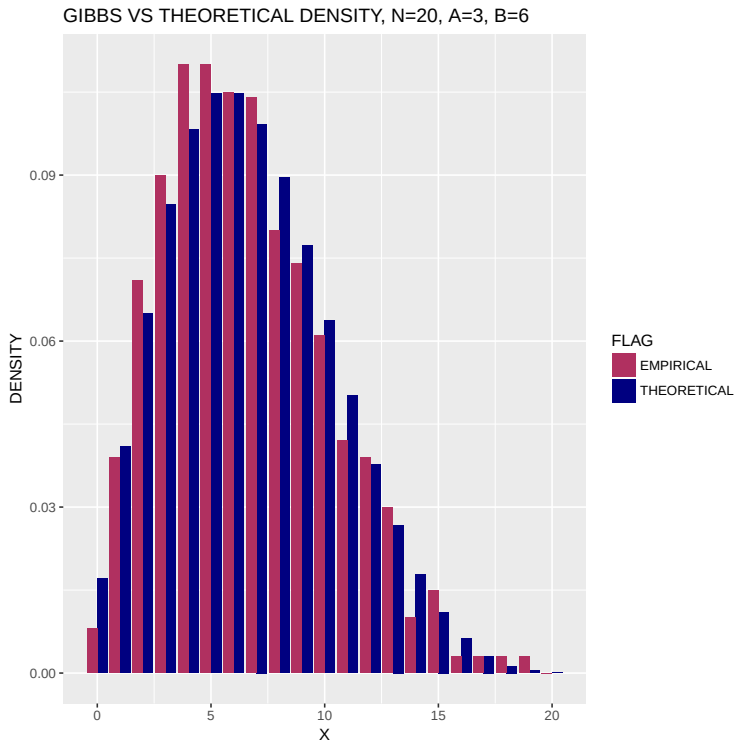
So I've tried different values of the parameters, different starting values of y_0, run larger Gibbs samples, and thinned the sample. At first I thought that my sample was not converging, but after looking at the results from part c. and re-running Gibbs with different parameters, I feel more confident. There doesn't seem to be any abnormal pattern to the sample, just that the values of X and Y seem uniformly dispersed among their supports (which is what caused my concern about a non-convergent sample).

part c.

To compare the Gibbs sample with a theoretical distribution, I'm going to take the empirical density of the Gibbs sample and then compare that to the theoretical density given by the betabinom.ab function in VGAM:

```
plot.df <- data.frame(X = rep(0:n, times = 2), FLAG = rep(c("EMPIRICAL",
    "THEORETICAL"), each = n + 1), DENS = numeric((n + 1) * 2))

gibbs <- results.df[, "X"]

for (i in 1:(n + 1)) {
    plot.df %<>% within({
        DENS[i] <- sum(gibbs == X[i])/(size/thin)
    })
}
```

```r
for (i in (n + 2):((n + 1) * 2)) {
    plot.df %<>% within({
        DENS[i] <- dbetabinom.ab(X[i], size = n, shape1 = a,
            shape2 = b)
    })
}

## Numerical evidence
(EX <- sum(0:n * dbetabinom.ab(0:n, size = n, shape1 = a, shape2 = b)))

## [1] 6.666667

mean(gibbs)

## [1] 6.61

## Graphical evidence
ggplot(plot.df, aes(x = X, y = DENS, fill = FLAG)) + geom_bar(stat = "identity",
    position = "dodge") + labs(title = sprintf("GIBBS VS THEORETICAL DENSITY, N=%d, A=%d, B=%d",
    n, a, b), y = "DENSITY") + scale_fill_manual(values = c("maroon",
    "navy"))
```



As I discussed in the second part, I was fairly sure the sample did not converge. The sample density doesn't exactly match the theoretical density. However, the mean of the sample and the expected value of X under the Beta-Binomial are strikingly close. So there does seem to be some degree of similarity between the Gibbs sample and reality. However, I'm still not 100% assured of the versimilitude of my sample.

3 Multinomial Model

part a.

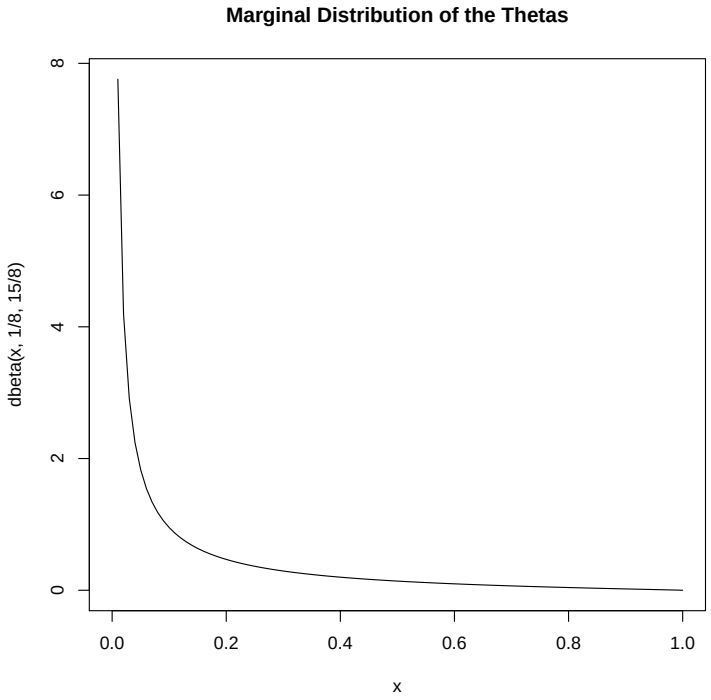
I'll assume that we are using the data from the paper as our data. The function of interest is Kappa.test in the fmsb package:

```
kappa.data <- matrix(c(5, 0, 0, 0, 3, 5, 0, 0, 0, 2, 3, 0, 0,
    0, 0, 3), ncol = 4, byrow = T)

fmsb::Kappa.test(kappa.data)

## $Result
##
##  Estimate Cohen's kappa statistics and test the null hypothesis
##  that the extent of agreement is same as random (kappa=0)
##
## data:  kappa.data
## Z = 5.0434, p-value = 2.287e-07
## 95 percent confidence interval:
##  0.4226336 0.9231608
## sample estimates:
## [1] 0.6728972
##
##
## $Judgement
## [1] "Substantial agreement"
```

part b.

Given a prior sample size of two and a "uniform" prior across the simplex, we say $\theta \sim D_{16}(\alpha)$, $\alpha = (1/8, \ldots, 1/8)^{\mathrm{T}}$. This results in a marginal of $\theta_i \sim \text{Beta}(1/8, 15/8)\ \forall i$, with expected value of $1/16$ for each θ_i. Here is a curve of the prior marginal for each θ_i:

```
curve(dbeta(x, 1/8, 15/8), from = 0, to = 1, main = "Marginal Distribution of the Thetas")
```



To calculate the Kappa value from the prior, we sample from the dirchlet distribution, then sample from the corresponding Multinomial sample with size 21. I then calculate the Kappa value on that Multinomial sample and take an average of all the Kappa values.

```
set.seed(12)
## So many closures.  What is this, javascript?
n <- 10000
(function(thetas) {

    out <- numeric(n)

    for (i in 1:n) {
        m <- matrix(rmultinom(1, 21, thetas[i, ]), ncol = 4,
            byrow = T)
        out[i] <- suppressWarnings(fmsb::Kappa.test(m)$Result$estimate)
    }

    mean(out, na.rm = T)

})(gtools::rdirichlet(n, rep(1/8, 16)))

## [1] 0.02051026
```

The estimate should be very close to zero, since the uniform Dirichlet prior would suggest "no agreement" between the two responders, as the responses are uniformly random and independent. Our Monte Carlo method seems to agree with this assessment.

part c.

The posterior distribution of $\theta|Y$ is a Dirichlet distribution with alpha values equal to the observed counts in kappa.data, plus 1/8. The predictive distribution of Y is Multinomial$(21, \theta)$. To calculate Kappa we sample from the posterior and then plug those in and sample from the above multinomial. We run the resultant matrix through Kappa.test and grab the point estimate and upper and lower bounds of the 95% interval. Then I take their means as the posterior estimates of the point estimate and confidence interval on Kappa.

```
(function(thetas) {

    out <- data.frame(low = numeric(n), high = numeric(n), estimate = numeric(n))

    for (i in 1:n) {
        m <- matrix(rmultinom(1, 21, thetas[i, ]), ncol = 4,
            byrow = T)
        r <- suppressWarnings(fmsb::Kappa.test(m)$Result)
        out[i, ] <- c(r$conf.int, r$estimate)
    }

    sapply(out, mean)

})(gtools::rdirichlet(n, kappa.data + 1/8))
```

```
##       low      high  estimate
## 0.3398424 0.8614670 0.6006547
```

This Bayesian analysis varies a lot from the frequentist analysis. First of all, the 95% confidence interval on Kappa is larger and shifted to the left in the Bayesian tack. The conclusion of "Substantial Agreement" holds for both analyses, however. The point estimates of Kappa are somewhat close: 0.67 in part a and 0.6 in part c.