MATH153, Lab 2

Senan Hogan-H.

15 February 2018

Completed with Jame Kinney. Work was about 50-50 in completion.

Question 1.

a. Need $\frac{\alpha}{\alpha+\beta} = .1$, so $9\alpha = \beta$.

$\Rightarrow$ want $pbeta(.2, \alpha, \beta) = .9$.

```
x <- 0.2
# a/(a+b) = 0.1, so b = 9a
vector <- seq(0, 10000, 0.001)
i <- 1
a <- vector[i]
b <- 9*a
epsilson <- 0.0001
while (abs(pbeta(x,a,b) - 0.9) > epsilson){
  i <- i + 1
  a <- vector[i]
  b <- 9*a
}
rm(vector, i)
a
```

```
## [1] 1.606
```

```
b
```

```
## [1] 14.45
```

```
pbeta(x,a,b) # roughly = 0.9
```

```
## [1] 0.8999
```

b. Need $\frac{\alpha}{\alpha+\beta} = mean$, so $\frac{\alpha(1-mean)}{mean} = \beta$.

$\Rightarrow$ want $pbeta(x, \alpha, \beta) = prob$.

```
ab_finder <- function(mean, x, prob){
  vector <- seq(0, 100, 0.001)
  i <- 1
  a <- vector[i]
  # a/(a+b) = mean
  b <- a*(1 - mean)*(1/mean)
  while (abs(pbeta(x,a,b) - prob) > 0.001){
    i <- i + 1
    a <- vector[i]
    b <- a*(1 - mean)*(1/mean)
  }
  rm(vector, i)
  return(c(a,b))
```

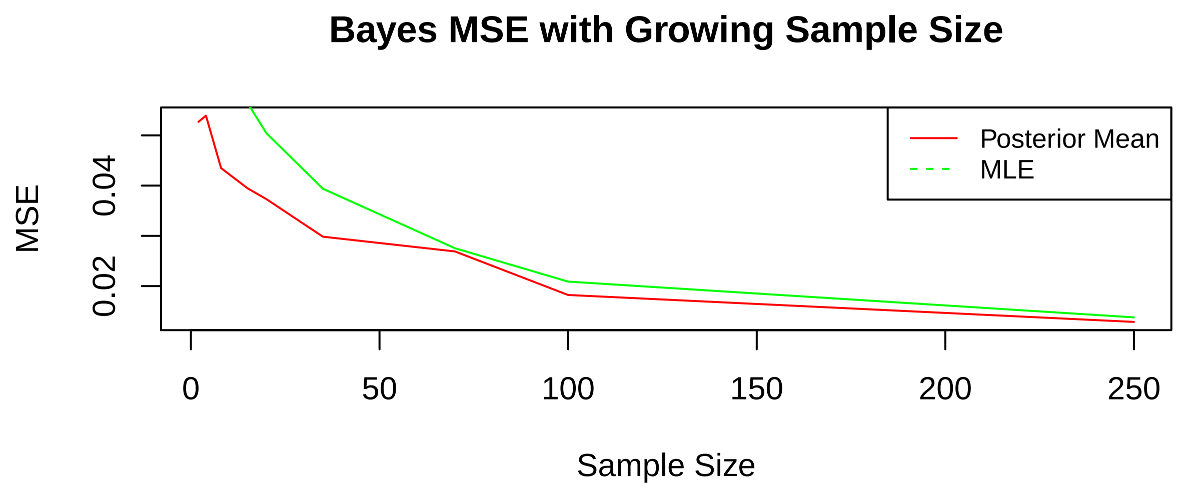
```r
}
ab_finder(0.1, 0.2, 0.9) # test the function.
```

```
## [1]  1.587 14.283
```

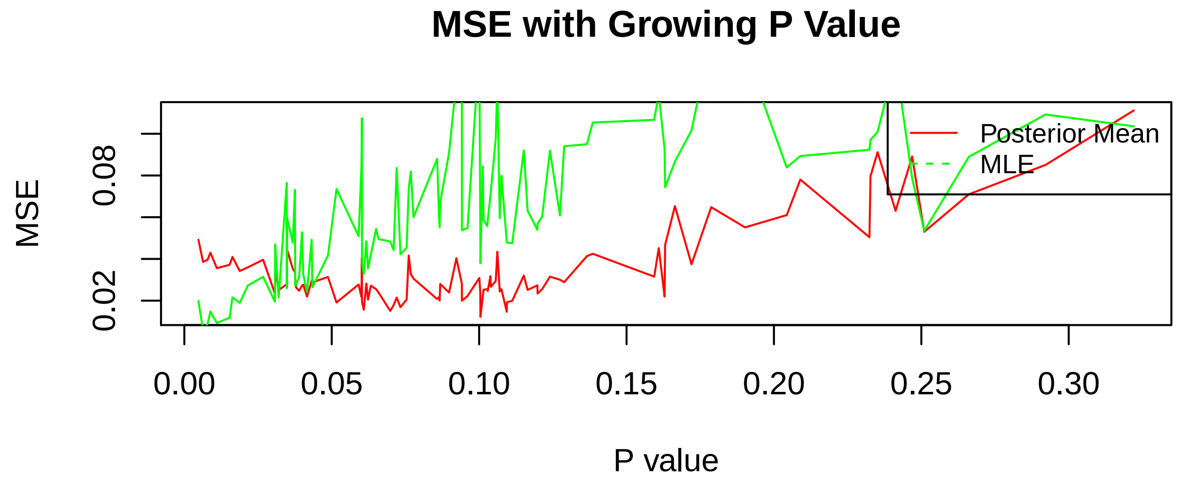
Question 2.

a.

```r
# Prior for p is beta (1.59, 14.31)\
alpha = 1.59
beta = 14.31
nP = 100 # Number of p's
nX = c(2, 4, 8, 15, 20, 35, 70, 100, 250) #Vector of sample sizes for the binomial experiments
sampleP = rbeta(nP, alpha, beta)
resultsMSE = matrix(nrow = (nP*length(nX)), ncol = 4)
colnames(resultsMSE) = c("mse_bayes", "mse_mle", "p", "n")
resultsMAE = matrix(nrow = (nP*length(nX)), ncol = 5)
colnames(resultsMAE) = c("mae_bayes_median", "mae_bayes_mean", "mae_mle", "p", "n")

j=1
for(i in 1:nP){
  for(n in nX){
    x = rbinom(1, n, sampleP[i])
    mse_mle = (sampleP[i]-x/n)^2
    mse_bayes = (sampleP[i]-(alpha+x)/(alpha+n+beta))^2
    mae_mle = abs(sampleP[i]-x/n)
    mae_bayes_mean = abs(sampleP[i]-(alpha+x)/(alpha+n+beta))
    mae_bayes_med = abs(sampleP[i]-(alpha+x-1/3)/(alpha+n+beta-2/3))
    resultsMSE[j,] = c(mae_bayes_mean, mae_mle, sampleP[i], n)
    resultsMAE[j,] = c(mae_bayes_med, mae_bayes_mean, mae_mle, sampleP[i], n)
    j=j+1
  }
}

bayesMeans_bySampleSize = c()
freqMeans_bySampleSize = c()
for(i in 1:length(nX)){
  bayesMeans_bySampleSize[i] = mean(resultsMSE[which(resultsMSE[,4]==nX[i]),1])
  freqMeans_bySampleSize[i] = mean(resultsMSE[which(resultsMSE[,4]==nX[i]),2])
}

sortedP = sort(sampleP)
bayesMeans_byP = c()
freqMeans_byP = c()
for(i in 1:length(sortedP)){
  bayesMeans_byP[i] = mean(resultsMSE[which(resultsMSE[,3]==sortedP[i]),1])
  freqMeans_byP[i] = mean(resultsMSE[which(resultsMSE[,3]==sortedP[i]),2])
}

plot(bayesMeans_bySampleSize~nX, main = "Bayes MSE with Growing Sample Size", xlab = "Sample Size", yla
lines(freqMeans_bySampleSize~nX, col = 'green')
legend("topright", legend=c("Posterior Mean", "MLE"), col=c("red", "green"), lty=1:2, cex=0.8)
```



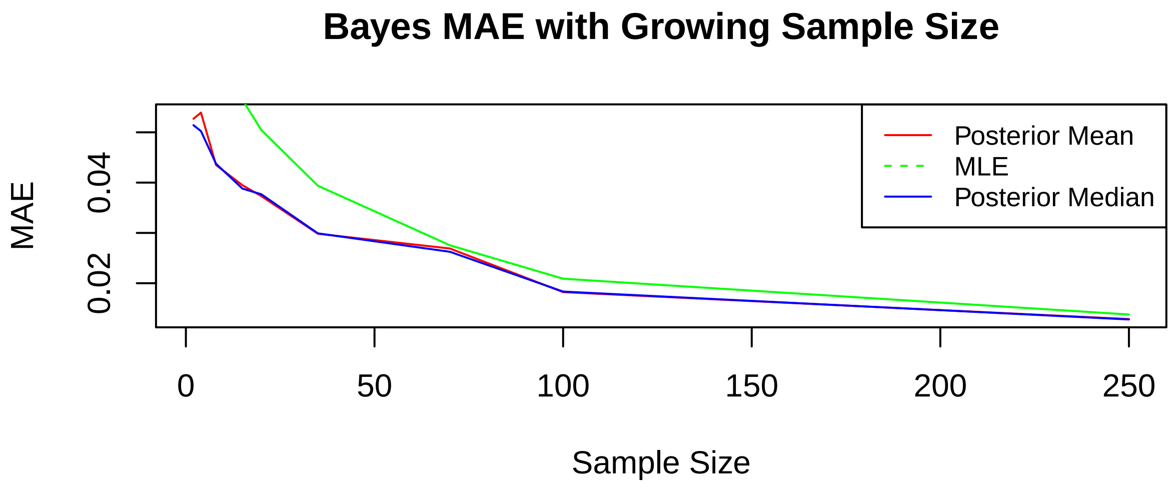
```
plot(bayesMeans_byP~sortedP, main = "MSE with Growing P Value", xlab = "P value", ylab = "MSE", col = '
lines(freqMeans_byP~sortedP, col = 'green')
legend("topright", legend=c("Posterior Mean", "MLE"), col=c("red", "green"), lty=1:2, cex=0.8)
```



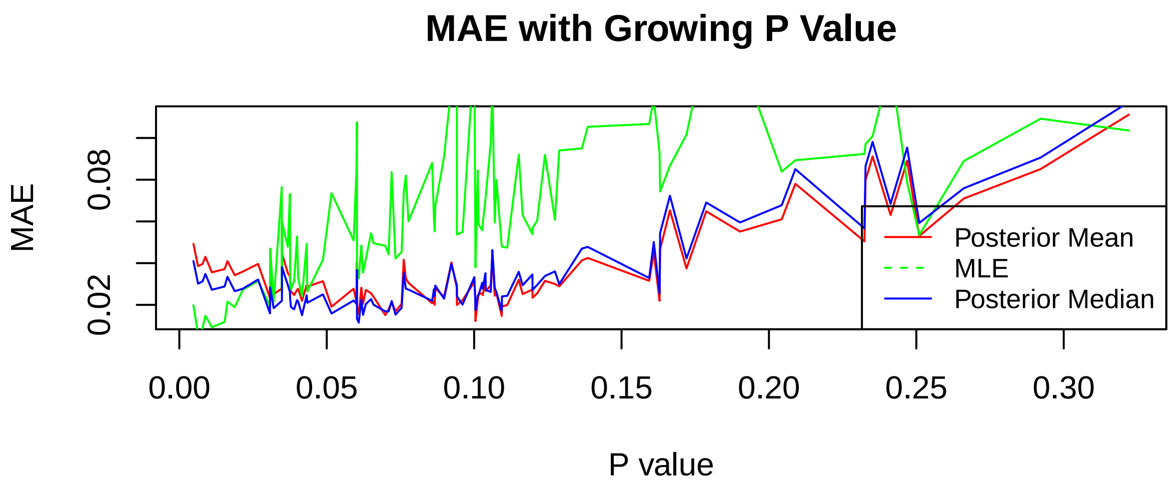
b.

```
bayesMeansMAE_bySampleSize = c()
bayesMedsMAE_bySampleSize = c()
freqMeansMAE_bySampleSize = c()
for(i in 1:length(nX)){
  bayesMedsMAE_bySampleSize[i] = mean(resultsMAE[which(resultsMAE[,5]==nX[i]),1])
  bayesMeansMAE_bySampleSize[i] = mean(resultsMAE[which(resultsMAE[,5]==nX[i]),2])
  freqMeansMAE_bySampleSize[i] = mean(resultsMAE[which(resultsMAE[,5]==nX[i]),3])
}

bayesMedsMAE_byP = c()
bayesMeansMAE_byP = c()
```

```
freqMeansMAE_byP = c()
for(i in 1:length(sortedP)){
  bayesMedsMAE_byP[i] = mean(resultsMAE[which(resultsMAE[,4]==sortedP[i]),1])
  bayesMeansMAE_byP[i] = mean(resultsMAE[which(resultsMAE[,4]==sortedP[i]),2])
  freqMeansMAE_byP[i] = mean(resultsMAE[which(resultsMAE[,4]==sortedP[i]),3])
}

plot(bayesMeansMAE_bySampleSize~nX, main = "Bayes MAE with Growing Sample Size", xlab = "Sample Size", y
lines(freqMeansMAE_bySampleSize~nX, col = 'green')
lines(bayesMedsMAE_bySampleSize~nX, col = 'blue')
legend("topright", legend=c("Posterior Mean", "MLE", "Posterior Median"), col=c("red", "green", "blue")
```



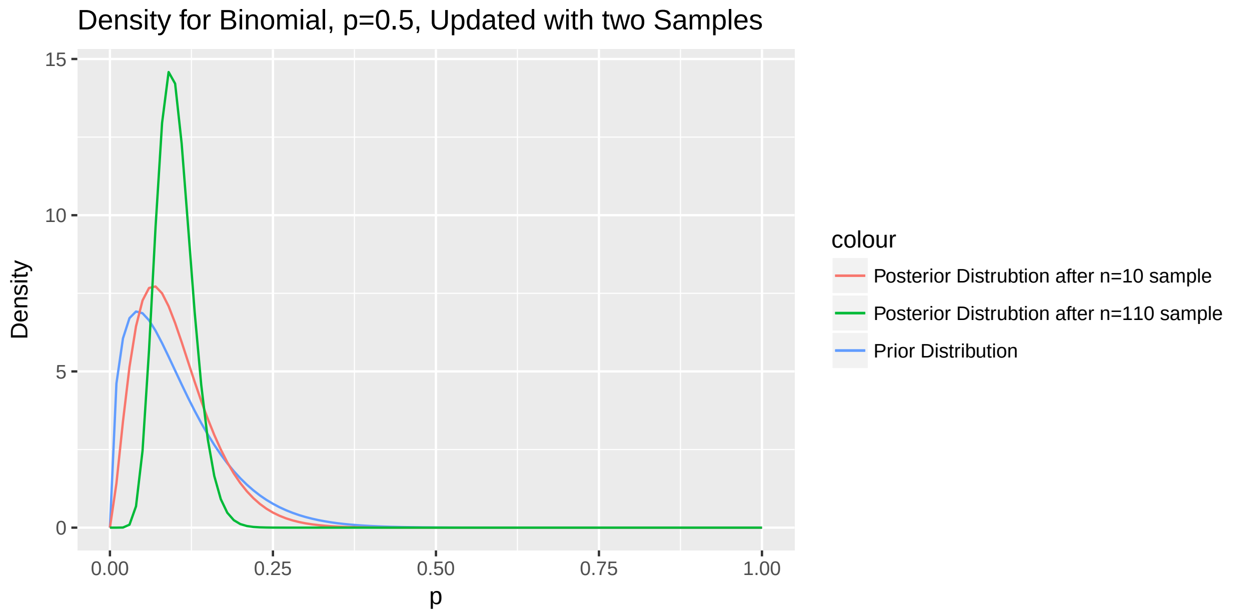
```
plot(bayesMeansMAE_byP~sortedP, main = "MAE with Growing P Value", xlab = "P value", ylab = "MAE", col =
lines(freqMeansMAE_byP~sortedP, col = 'green')
lines(bayesMedsMAE_byP~sortedP, col = 'blue')
legend("bottomright", legend=c("Posterior Mean", "MLE", "Posterior Median"), col=c("red", "green", "blu
```



Question 3.

a. Start with an informative and good prior, as outlined in question 1 for binomial of $p = 0.1$. In this example suppose that the correct estimate is $p = 0.1$ (thus we have a good prior), so that $\frac{1}{n}\sum_{i=1}^{n} X_i = \frac{X}{n} = 0.1$ for the (fixed) MLE estimator.

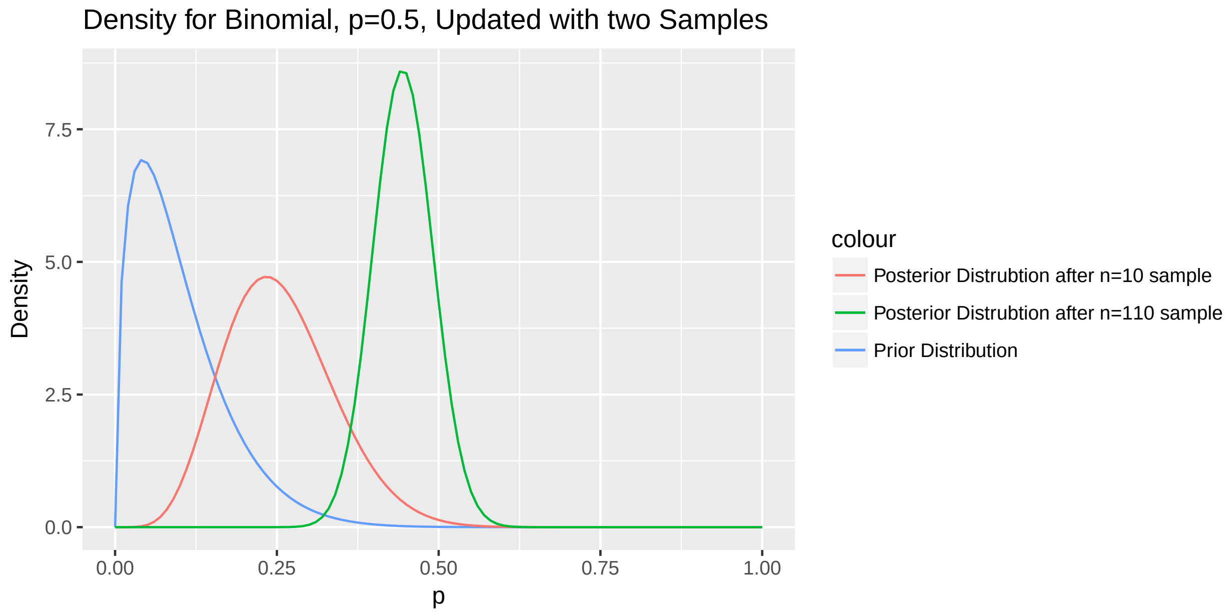
```r
p <- 0.1
alpha <- ab_finder(0.1, 0.2, 0.9)[1]
beta <- ab_finder(0.1, 0.2, 0.9)[2]

prob <- seq(0,1,.01)
# First
x_0 <- dbeta(prob, alpha, beta)

# Update with data sample of 10
n <- 10
x <- p*n
x_10 <- dbeta(prob, alpha + x, beta + n - x)

# Update with another data sample of 100
n <- 100
x <- p*n
x_100 <- dbeta(prob, alpha + x, beta + n - x)

data.frame(x_0, x_10, x_100, prob) %>% ggplot(aes(x= prob)) +
  geom_line(aes(y= x_0, colour = 'Prior Distribution')) +
  geom_line(aes(y= x_10, colour = 'Posterior Distrubtion after n=10 sample')) +
  geom_line(aes(y= x_100, colour = 'Posterior Distrubtion after n=110 sample')) +
  labs(title = "Density for Binomial, p=0.5, Updated with two Samples",
       x = "p", y = "Density")
```



The prior distribution is centred around the correct value for p, and is a good estimate for the distribution of $\hat{p}$. Updating ths distribution with data samples focus the distribution around the correct value, increasing density for the correct value even further.

b. Start with an informative yet bad prior, as outlined in question 1 for binomial of $p = 0.1$. In this example suppose that the correct estimate is however $p = 0.5$, so that $\frac{1}{n}\sum_{i=1}^{n} X_i = \frac{X}{n} = 0.5$ for the (fixed) MLE estimator.

```
p <- 0.5
alpha <- ab_finder(0.1, 0.2, 0.9)[1]
beta <- ab_finder(0.1, 0.2, 0.9)[2]

prob <- seq(0,1,.01)
# First
x_0 <- dbeta(prob, alpha, beta)

# Update with data sample of 10
n <- 10
x <- p*n
x_10 <- dbeta(prob, alpha + x, beta + n - x)

# Update with another data sample of 100
n <- 100
x <- p*n
x_100 <- dbeta(prob, alpha + x, beta + n - x)

data.frame(x_0, x_10, x_100, prob) %>% ggplot(aes(x= prob)) +
  geom_line(aes(y= x_0, colour = 'Prior Distribution')) +
  geom_line(aes(y= x_10, colour = 'Posterior Distrubtion after n=10 sample')) +
  geom_line(aes(y= x_100, colour = 'Posterior Distrubtion after n=110 sample')) +
  labs(title = "Density for Binomial, p=0.5, Updated with two Samples",
       x = "p", y = "Density")
```



```
rm(list=ls())
```

The prior distribution is centred below 0.1, as our prior estimates $p = 0.1$. However, after observation of great data points, the posterior distribution better fits the *real* distribution of $p = 0.5$, so that with greater sample size the distribution shifts from the original estimate to the correct estimate.