MATH153, Lab 1

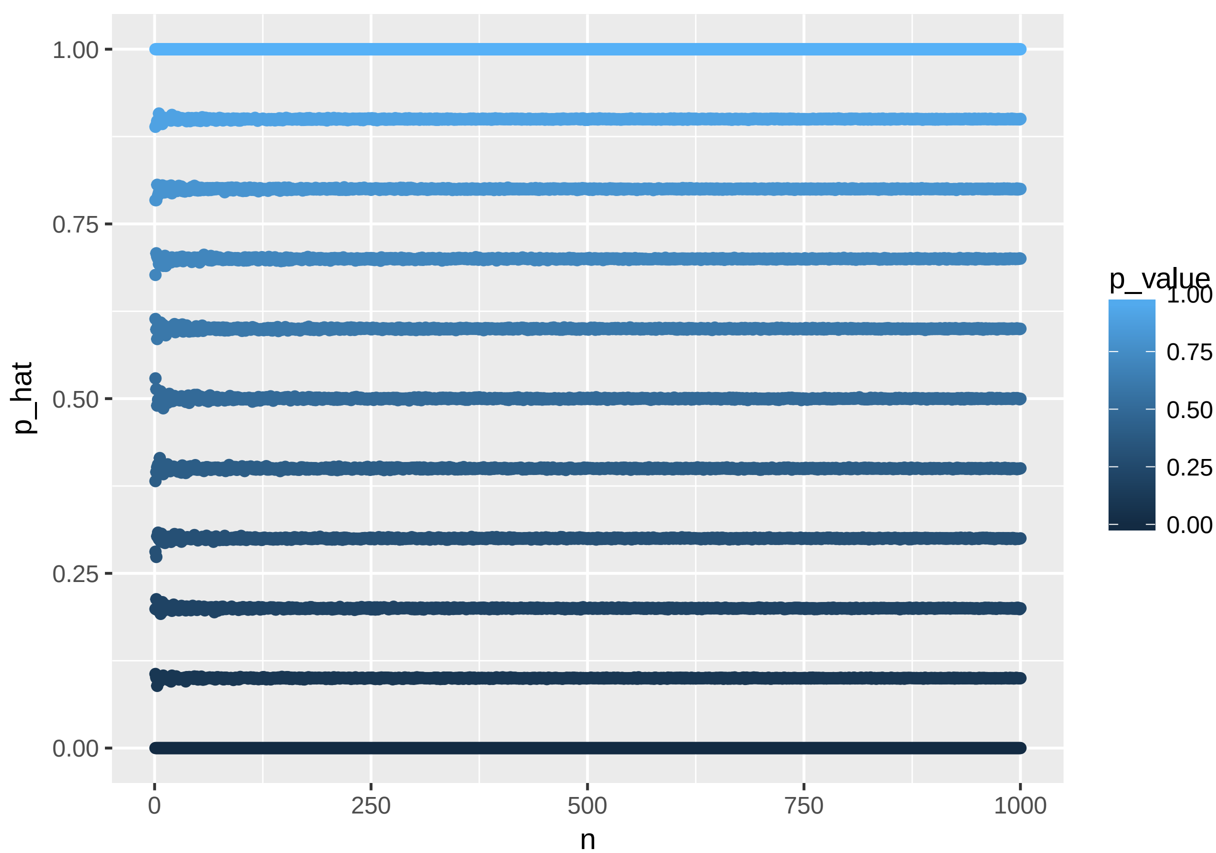
Senan Hogan-H.

25 January 25 2018

Question 1.

a.

```
p_hat <- c() #initialize the vector that I will fill in with the partial sums
p_value <- c()
p = c(0:10)/10 # Vector of p values to try over
n = c(1:1000) #Vector of sample size to repeat over
for (k in p) {
  for (i in n) {
    p_i <- mean(rbinom(1000, i, k))/i
    p_hat <- c(p_hat, p_i)
    p_value <- c(p_value, k)
  }
}
P.data <- data.frame(n, p_hat, p_value)
P.data %>% ggplot(aes(n, p_hat, col=p_value)) +
                    geom_point() + ylim(0, 1)
```



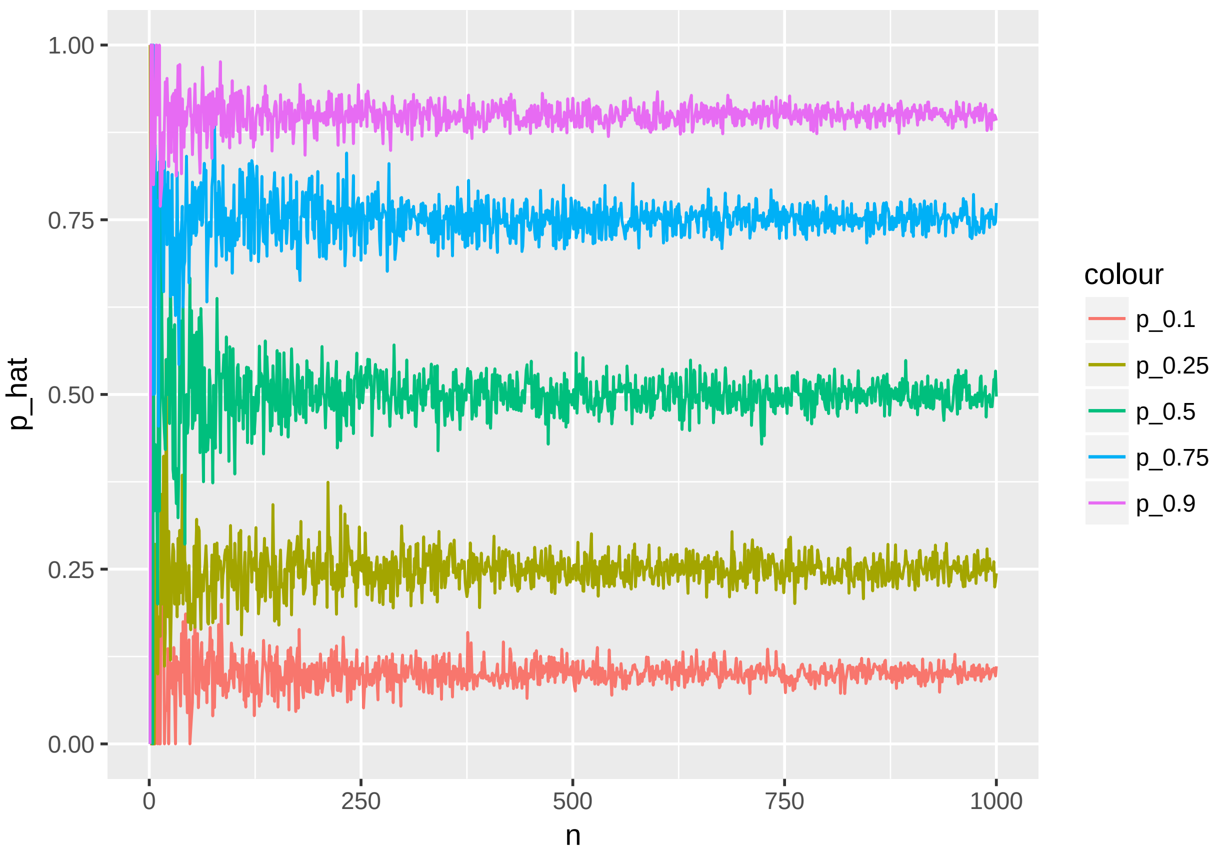
$$E[\hat{p}] = E[\frac{\sum_{i=1}^{n} X_i}{n}] = \frac{1}{n}\sum_{i=1}^{n} X_i = \frac{1}{n}pn = p$$

. So that $\hat{p}$ is unbiased in general, and so sample size does not matter for this small-sample property.

b.

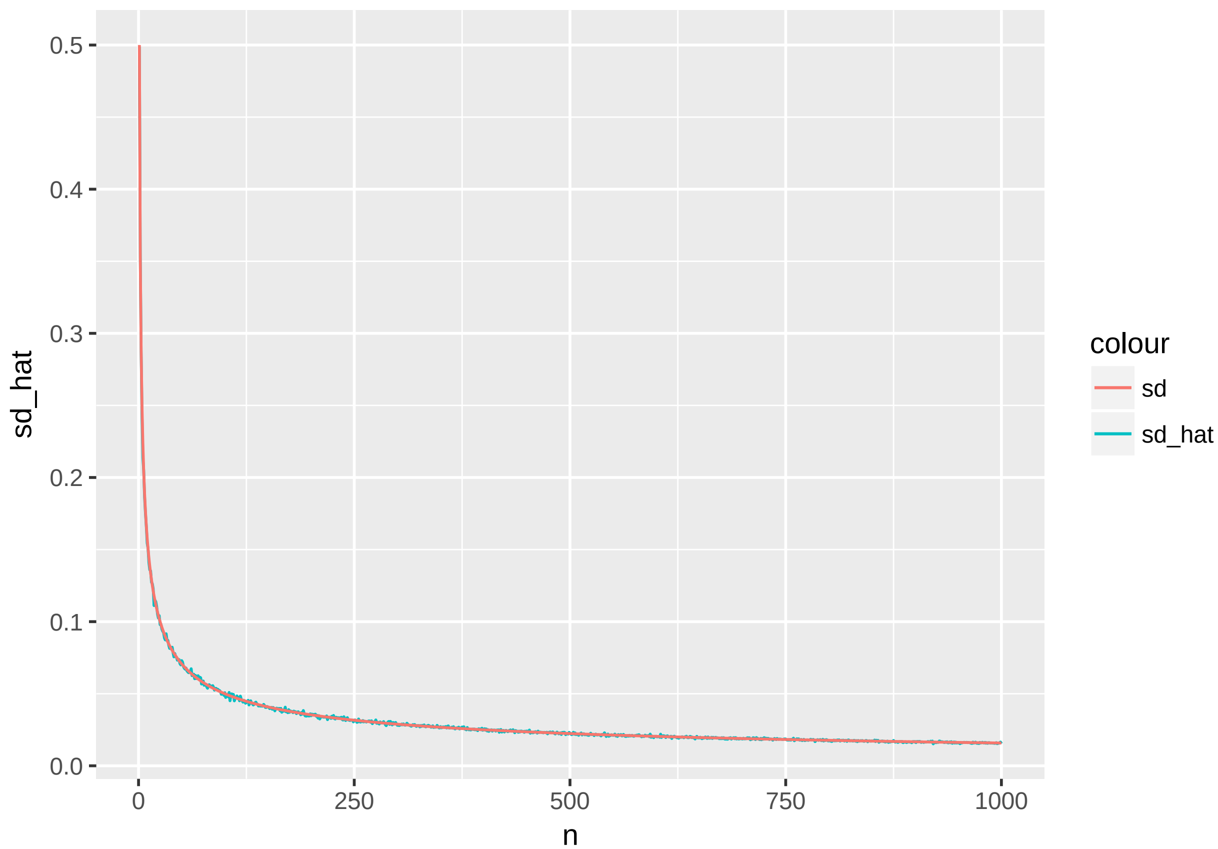
```
p_i <- p_0.1 <- p_0.25 <- p_0.5 <- p_0.75 <- p_0.9 <- c()
n = c(1:1000) #Vector of sample size to repeat over
for (i in n) {
  p_i <- mean(rbinom(1, i, 0.1))/i
  p_0.1 <- c(p_0.1, p_i)
}
for (i in n) {
  p_i <- mean(rbinom(1, i, 0.25))/i
  p_0.25 <- c(p_0.25, p_i)
}
for (i in n) {
  p_i <- mean(rbinom(1, i, 0.5))/i
  p_0.5 <- c(p_0.5, p_i)
}
for (i in n) {
  p_i <- mean(rbinom(1, i, 0.75))/i
  p_0.75 <- c(p_0.75, p_i)
}
for (i in n) {
  p_i <- mean(rbinom(1, i, 0.9))/i
  p_0.9 <- c(p_0.9, p_i)
}

P.data <- data.frame(n, p_0.1 , p_0.25 , p_0.5 , p_0.75 , p_0.9)
P.data %>% ggplot(aes(n)) +
  geom_line(aes(y = p_0.1, colour = "p_0.1")) +
  geom_line(aes(y = p_0.25, colour = "p_0.25")) +
  geom_line(aes(y = p_0.5, colour = "p_0.5")) +
  geom_line(aes(y = p_0.75, colour = "p_0.75")) +
  geom_line(aes(y = p_0.9, colour = "p_0.9")) +
  ylim(0, 1) + ylab('p_hat')
```



Question 2.

```
p=0.5
n=c(1:1000)
sd_hat=c()
sd = c()
for (i in n) {
  x = rbinom(1000,i,p)
  sd_hat=c(sd_hat,sd(x)/i)
  sd = c(sd, p/sqrt(n))
}
P.data <- data.frame(n, sd_hat, sd)
P.data %>%
  ggplot(aes(x=n)) +
  geom_line(aes(y=sd_hat, colour = 'sd_hat')) +
  geom_line(aes(y=sd, colour = 'sd'))
```



Question 3.

```
#Initialise vectors for table output
Observations <- p_value <- Bayesian <- MLE <- c()
#First p-value
for (p in c(1:10)/10){
  n <- 10
  p_hatb <- p_hat <- c()
  for (i in c(1:10000)){
    x_i <- rbinom(n, 1, p)
    p_hatb <- c((sum(x_i)+10)/(n+20), p_hatb)
    p_hat <- c(mean(x_i), p_hat)
  }
  Observations <- c(n, Observations)
  p_value <- c(p, p_value)
  Bayesian <- c(mean((p_hatb - p)^2), Bayesian)
  MLE <- c(mean((p_hat - p)^2), MLE)
  #second n value
  n <- 100
  p_hatb <- p_hat <- c()
  for (i in c(1:10000)){
    x_i <- rbinom(n, 1, p)
    p_hatb <- c((sum(x_i)+10)/(n+20), p_hatb)
    p_hat <- c(mean(x_i), p_hat)
```

```r
  }
  Observations <- c(n, Observations)
  p_value <- c(p, p_value)
  Bayesian <- c(mean((p_hatb - p)^2), Bayesian)
  MLE <- c(mean((p_hat - p)^2), MLE)
}
P.data <- data.frame(p_value, Observations, Bayesian, MLE)
tab <- xtable(P.data)
print(tab, type="latex")
```

```
## \begin{tabular}{rrrrr}
##   \hline
##  & p\_value & Observations & Bayesian & MLE \\
##   \hline
## 1 & 1.00 & 100.00 & 0.01 & 0.00 \\
##   2 & 1.00 & 10.00 & 0.11 & 0.00 \\
##   3 & 0.90 & 100.00 & 0.01 & 0.00 \\
##   4 & 0.90 & 10.00 & 0.07 & 0.01 \\
##   5 & 0.80 & 100.00 & 0.00 & 0.00 \\
##   6 & 0.80 & 10.00 & 0.04 & 0.02 \\
##   7 & 0.70 & 100.00 & 0.00 & 0.00 \\
##   8 & 0.70 & 10.00 & 0.02 & 0.02 \\
##   9 & 0.60 & 100.00 & 0.00 & 0.00 \\
##   10 & 0.60 & 10.00 & 0.01 & 0.02 \\
##   11 & 0.50 & 100.00 & 0.00 & 0.00 \\
##   12 & 0.50 & 10.00 & 0.00 & 0.02 \\
##   13 & 0.40 & 100.00 & 0.00 & 0.00 \\
##   14 & 0.40 & 10.00 & 0.01 & 0.02 \\
##   15 & 0.30 & 100.00 & 0.00 & 0.00 \\
##   16 & 0.30 & 10.00 & 0.02 & 0.02 \\
##   17 & 0.20 & 100.00 & 0.00 & 0.00 \\
##   18 & 0.20 & 10.00 & 0.04 & 0.02 \\
##   19 & 0.10 & 100.00 & 0.01 & 0.00 \\
##   20 & 0.10 & 10.00 & 0.07 & 0.01 \\
##    \hline
## \end{tabular}
```

For non-extreme (i.e. close to 0.5) p-values, the Bayesian estimator has a lower variance than does the MLE estimator.