Lab 1

Harry Bendekgey (and Peter Brody-Moore)
Math 153: Bayesian Statistics

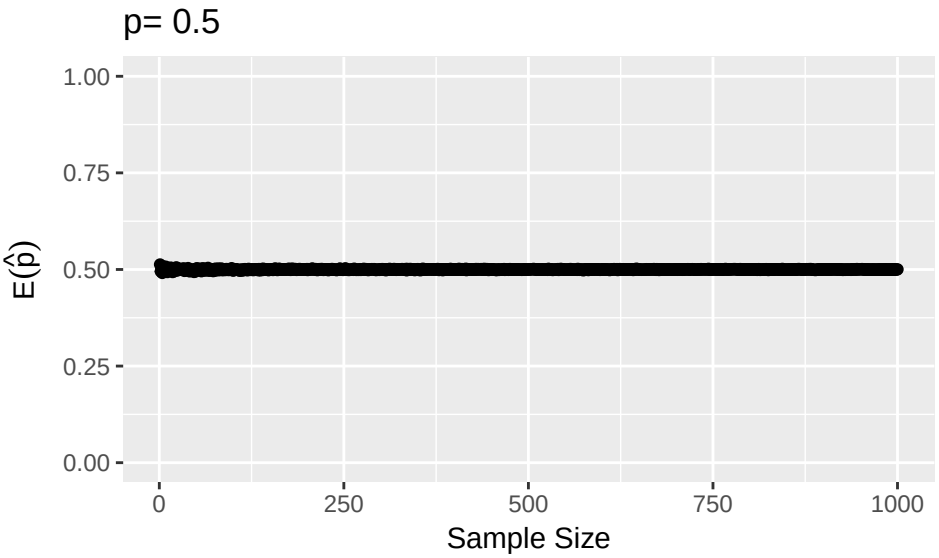
January 25, 2018

Problem 1

Analyzing $E(\hat{p})$

We begin by imagining a simple binomial experiment. We want to visualize $E(\hat{p})$ at various sample sizes. We set $p = 0.5$ and for every sample size between 1 and 1000 estimate $E(\hat{p})$:

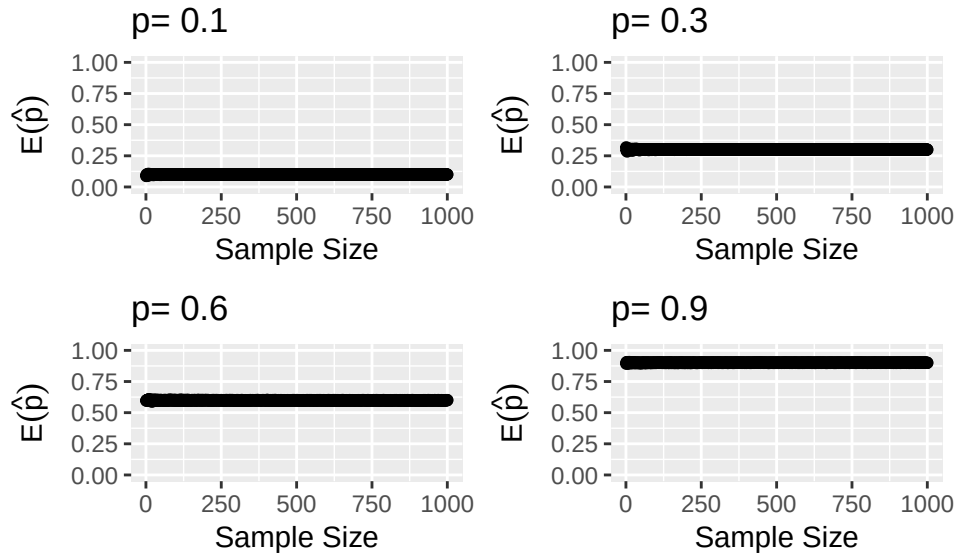
```
sample_sizes <- 1:1000
graph_means <- function(p) {
  means <- c()
  for (i in 1:length(sample_sizes)) {
    means[i] <- mean(rbinom(1000, sample_sizes[i], p) / sample_sizes[i])
  }
  df <- data.frame(sample_sizes,means)
  return(ggplot(df, aes(sample_sizes, means)) +
           geom_point() +
           scale_y_continuous(limits = c(0,1)) +
           labs(x = "Sample Size", y = expression(E(hat(p))), title = paste("p=",p)))
}

graph_means(0.5)
```



We can see that $E(\hat{p})$ is constant, and that even at the smallest sample sizes, it takes the value 0.5. We claim that $E(\hat{p}) = p$. This reflects the property we learned in class: $E(\hat{\theta}_{MLE}) \to \theta$.

To confirm our findings, we will run a similar experiment using other values for p:

```
ggarrange(graph_means(0.1), graph_means(0.3),
          graph_means(0.6), graph_means(0.9), ncol=2, nrow=2)
```



This confirms our conjecture that $E(\hat{p}) = p$ is constant.

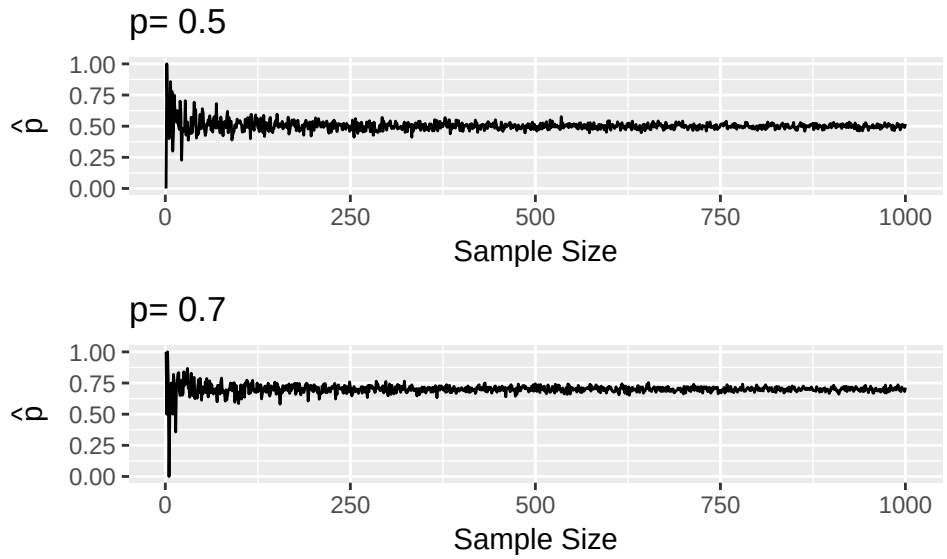
We can also prove this relationship mathematically:

$$E(\hat{p}) = E\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n}\sum_{i=1}^{n} E(X_i) = \frac{1}{n}\sum_{i=1}^{n} p = \frac{1}{n}(np) = p$$

Analyzing $\hat{p}$

We want to show that $\hat{p}$ converges to p as n gets larger. We graph a single $\hat{p}$ value for each sample size between 1 and 1000:

```
graph_phats <- function(p) {
  p_hats <- c()
  for (i in 1:length(sample_sizes)) {
    p_hats[i] <- rbinom(1, sample_sizes[i], p) / sample_sizes[i]
  }
  df <- data.frame(sample_sizes,p_hats)
  return(ggplot(df, aes(sample_sizes, p_hats)) +
      geom_line() +
      scale_y_continuous(limits = c(0,1)) +
      labs(x = "Sample Size", y = expression(hat(p)), title = paste("p=",p)))
}

ggarrange(graph_phats(0.5),graph_phats(0.7),ncol=1, nrow=2)
```



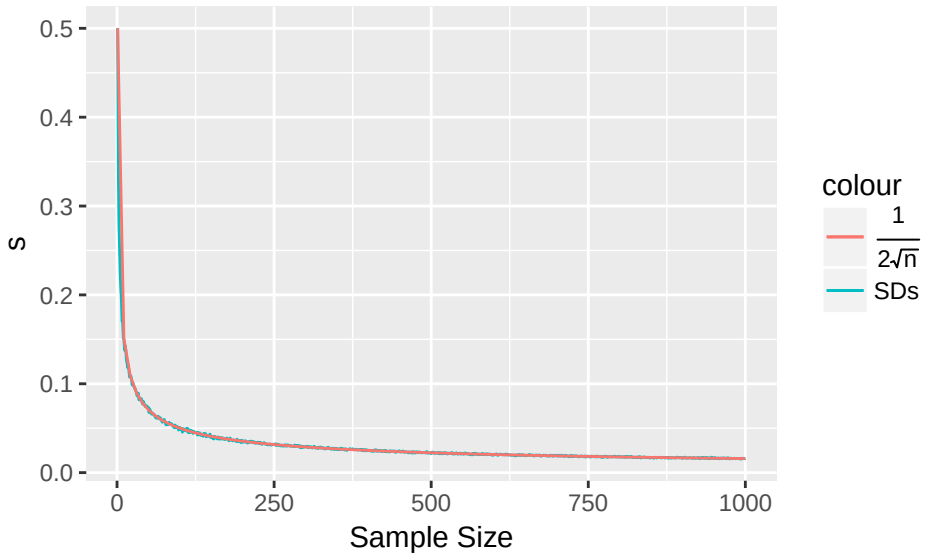
These results match our expectation: as the sample size gets larger, $\hat{p}$ converges to p.

Problem 2

Analyzing $s_{\hat{p}}$

Using the Central Limit Theorem, we approximate that $\hat{p} \sim N(p, \sqrt{\frac{p(1-p)}{n}}^2)$. In the case of $p = 0.5$ we expect the standard deviation to be $\sqrt{\frac{0.5^2}{n}} = \frac{1}{2\sqrt{n}}$. We can confirm that this is true by sampling many $\hat{p}$ values at different sample sizes and see that it behaves like the equation $f(x) = \frac{1}{2\sqrt{n}}$

```
sds <- c()
for (i in 1:length(sample_sizes)) {
  sds[i] <- sd(rbinom(1000, sample_sizes[i], 0.5) / sample_sizes[i])
}
df <- data.frame(sample_sizes,sds)
fun.1 <- function(x) 1/(2 * sqrt(x))
ggplot(df) +
    geom_line(aes(sample_sizes, sds, color="red")) +
    stat_function(fun = fun.1, aes(color="blue")) +
    scale_color_discrete(labels = c(expression(frac(1,2 * sqrt(n))), "SDs")) +
    labs(x = "Sample Size", y = "s")
```



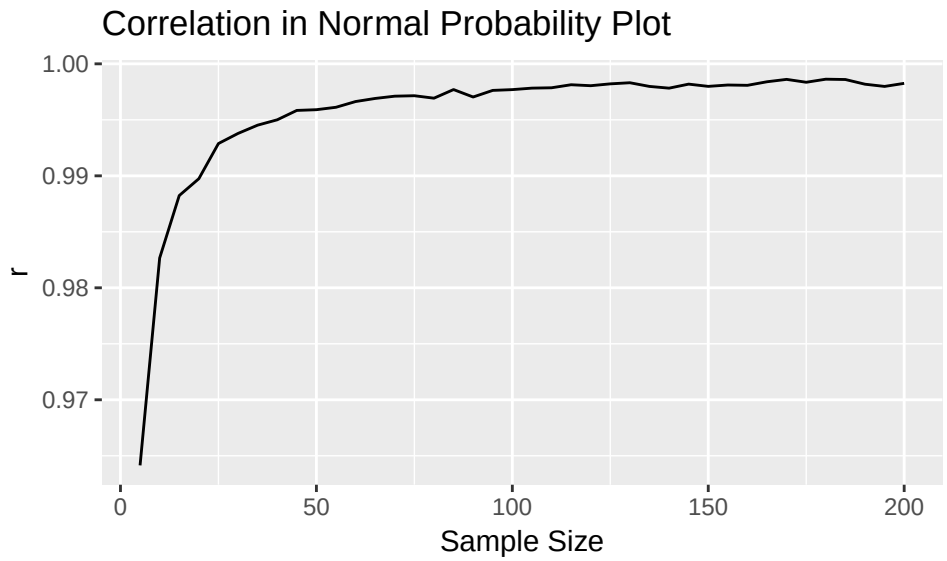
Here we can see that they are almost identical, supporting our claim.

Analyzing the distribution of $\hat{p}$

We're also interested in if the $\hat{p}$ values are distributed normally. We do this by constructing normal probability plots. This is done by ordering the data and comparing $\hat{p}_i$ to $E(\hat{p}_i)$ where $\hat{p}_i$ is the i^{th} smallest $\hat{p}$ from the distribution. If the data is normally distributed, we expect a linear normal probability plot. To test this hypothesis, we create a normal probability plot for each sample size of interest and calculate their correlation.

```
n <- seq(5,200,5)
cor_npp <- c()
for (i in 1:length(n)) {
  p_hats <- rbinom(1000, n[i], 0.5) / n[i]
  npp <- qqnorm(p_hats)
  cor_npp[i] <- cor(npp$x, npp$y)
}
```

```
df <- data.frame(n, cor_npp)
ggplot(df, aes(n, cor_npp)) +
      geom_line() +
      labs(x = "Sample Size", y = expression(r), title = "Correlation in Normal Probability Plot")
```



Here we can see that the distribution of $\hat{p}$-values assymptotically approaches normality as the sample size gets bigger.

Problem 3

Comparing Bayesian Point Estimators to Maximum Likelihood Estimators

```
bayesian <- function(X){
  phat <- (sum(X)+10)/(length(X)+20)
  return(phat)
}
frequentist <- function(X){
  phat <- sum(X)/length(X)
  return(phat)
}
genX <- function(n,p){
  X <- rbinom(n=n,size=1,prob=p)
  return(X)
}
genBPE <- function(n,p){
  X <- genX(n,p)
  bpe <- bayesian(X)
  return(bpe)
}
genMLE <- function(n,p){
  X <- genX(n,p)
  mle <- frequentist(X)
  return(mle)
}
bpeMSE <- function(n,p){
  phats <- replicate(10000,genBPE(n,p)) # generates 10000 phats
  mse <- mean((phats-p)^2)
```

```
  return(mse)
}
mleVar <- function(n,p){
  variance <- var(replicate(10000,genMLE(n,p))) # generates 10000 phats
  return(variance)
}

num <- c(10,100)
prob <- c(.25,.5,.75)

compare <- NULL # initialize dataframe
for (p in prob){ # loop through all n's and p's.
  for (n in num){
    mle_var <- mleVar(n,p)
    bpe_var <- bpeMSE(n,p)
    compare = rbind(compare, data.frame(n, p, mle_var, bpe_var))
  }
}
compare
```

```
##     n    p mle_var bpe_var
## 1  10 0.25 0.01867 0.02994
## 2 100 0.25 0.00184 0.00304
## 3  10 0.50 0.02541 0.00274
## 4 100 0.50 0.00248 0.00170
## 5  10 0.75 0.01851 0.03006
## 6 100 0.75 0.00184 0.00301
```

To compare the Bayesian Point Estimate to the Maximum Likelihood Estimator, we use the mean squared error of 10,000 randomnly generated $\hat{p}$'s for the BPE and the variance of 10,000 randomnly generated $\hat{p}$'s for the MLE. We compared these two estimators across different combinations of n and p. When looking at the data in the above table, it is clear that the BPE performs much better than the MLE when $p = .5$. This is likely because the BPE adds 10 to the numerator and 20 to the denominator, pushing our estimator towards .5. Similarly, the MLE performs better for $p = .25$ and $p = .75$ because the BPE is pushing away from these values. So, the BPE is preferable if the true value of p is close to .5, and the MLE is preferable otherwise for this given scenario.