Multilevel Modeling

Answer Key: Homework 1

Andy Stone

2.2

a.

The actual standard deviation is found by:

$$s = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}} = 0.0064$$

The expected standard deviation is found by:

$$sd^{expected} = \sqrt{\hat{p}(1-\hat{p})/n} = 0.008$$

```
births <- c(.4777, .4875, .4859, .4754, .4874, .4864, .4813, .4787, .4895, .4797, .4876,
            .4859, .4857, .4907, .5010, .4903, .4860, .4911, .4871, .4725, .4822, .4870,
            .4823, .4973)

# Standard deviation of the proportions
sd(births)
[1] 0.006409724

# Expected standard deviation
(expected <- sqrt(mean(births) * (1 - mean(births)) * (1/3900)))
[1] 0.008003121
```

b.

To determine if the expected and actual standard deviations are statistically distinguishable, we can use a two-sided chi-square test. This has the associated test statistic:

$$T = \frac{(N-1) * s^2}{\sigma^2}$$

```
# Test statistic
print(chi.squared.statistic <- (23 * (0.0064)^2)/(0.008^2))
[1] 14.72

# Critical values
c(qchisq(0.025, 23), qchisq(0.975, 23))
[1] 11.68855 38.07563
```

The test statistic is contained within the critical values, so the difference is not statistically significant.

2.3

The normal density curve is very similar to the histogram. This is to be expected, as a result of the Central Limit Theorem. The CLT says that, no matter the underlying distribution, the sum of many independent random variables (from a well-behaved distribution) will be an approximately normally distributed random variable.

```
set.seed(100) # For replication
xvals <- NULL # Empty vector to add simulated values of x into

# Creating the 1000 simulations of x
for(i in 1:1000){
  xvals <- c(xvals, sum(runif(20, min=0, max=1)))
}

par(mfrow=c(1,1)) # Plotting parameters
h <- hist(xvals, plot=FALSE) # Characteristics of the histogram
ylim1 <- range(0, h$density, dnorm(0)) # Setting range of y values for histogram

# Plotting histogram
hist(xvals, freq=FALSE, ylim=ylim1, main="Figure 1: Histogram of Simulated x Values
     and Normal Density Curve", xlab="x Values", cex.lab=0.7, cex.main=0.9, cex.axis=0.7)
# Overplotting normal density
curve(dnorm(x, mean=mean(xvals), sd=sd(xvals)), add=TRUE, col="red")
```