Homework 1

Neal Marquez

September 28, 2018

Homework 1

Question 1

The fraction of males in a given population is 0.49. A) What is the probability of sampling 22 males in a random sample of size 30? How about sampling 16 males in a random sample of size 30? Calculate your answers by hand. B) Next, verify your results using R's built-in functions for calculating probabilities. C) Finally, write R code to simulate each of these scenarios many times, and report the results of your simulations.

A

Using the function below we can calculate the probability of observing k males in a sample of n individuals if the probability of selecting a male in the total population is p. In our test case the n is always 30, p is always .49, and k is either 22 or 16. The function follows the formula presented below.

```
handBinom <- function(k,n,p){
    factorial(n) / (factorial(k) * factorial(n-k)) * p^k * (1-p)^(n-k)
}
```

$$\Pr(k|n,p) = \underbrace{\binom{n}{k}}_{\text{\# of ways to get k}} \times \underbrace{\prod f_{\text{Bern}}(x_{ij}|p)}_{\text{Pr(getting k)}}$$

$$\Pr(k|n,p) = \frac{n!}{k!(n-k)!}p^k(1-p)^{n-k}$$

$$\Pr(22|30,.49) = \frac{30!}{22!(30-22)!}.49^{22}(1-.49)^{30-22}$$

$$\Pr(22|30,.49) \approx$$

```
cat(round(handBinom(22, 30, .49), 6))
```

```
## 0.004095
```

$$\Pr(16|30,.49) = \frac{30!}{16!(30-16)!}.49^{16}(1-.49)^{30-16}\Pr(16|30,.49) \approx$$

```
cat(round(handBinom(16, 30, .49), 6))
```

```
## 0.129346
```

B

Using the `R` built in functions we find that we get the same results.

```
cat(round(dbinom(22, size=30, prob=.49), digits=6))
```

```
## 0.004095
```

```
cat(round(dbinom(16, size=30, prob=.49), digits=6))
```

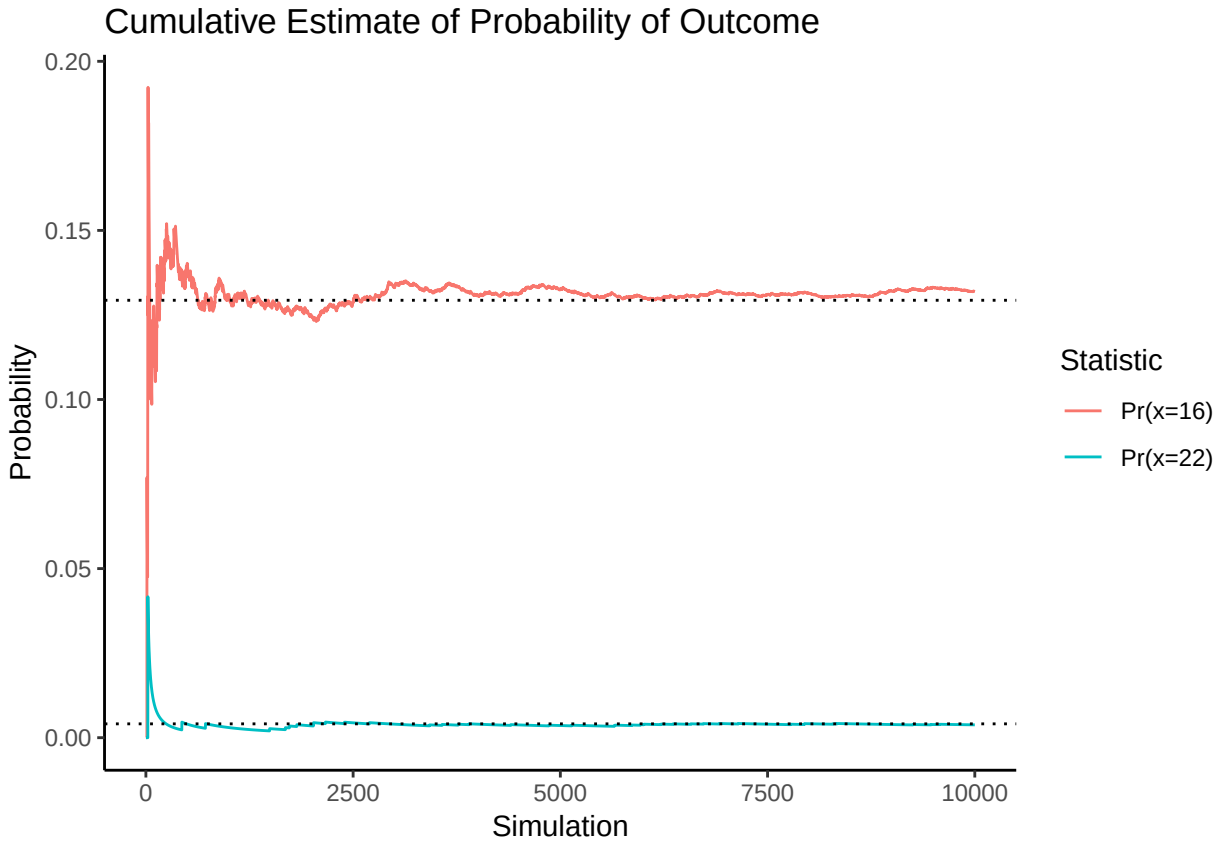
```
## 0.129346
```

C

We can try and simulate the values in a few different ways in order to get at the true underlying probability. One way is to take a set of n simulations from the Binomial distribution each with 30 flips and indicate each time an observation equaled either 22 or 16 with a 1 in two separate vectors. We then can take the cumulative mean for each of these vectors and see how the means converge to the truth as the sample size increases.

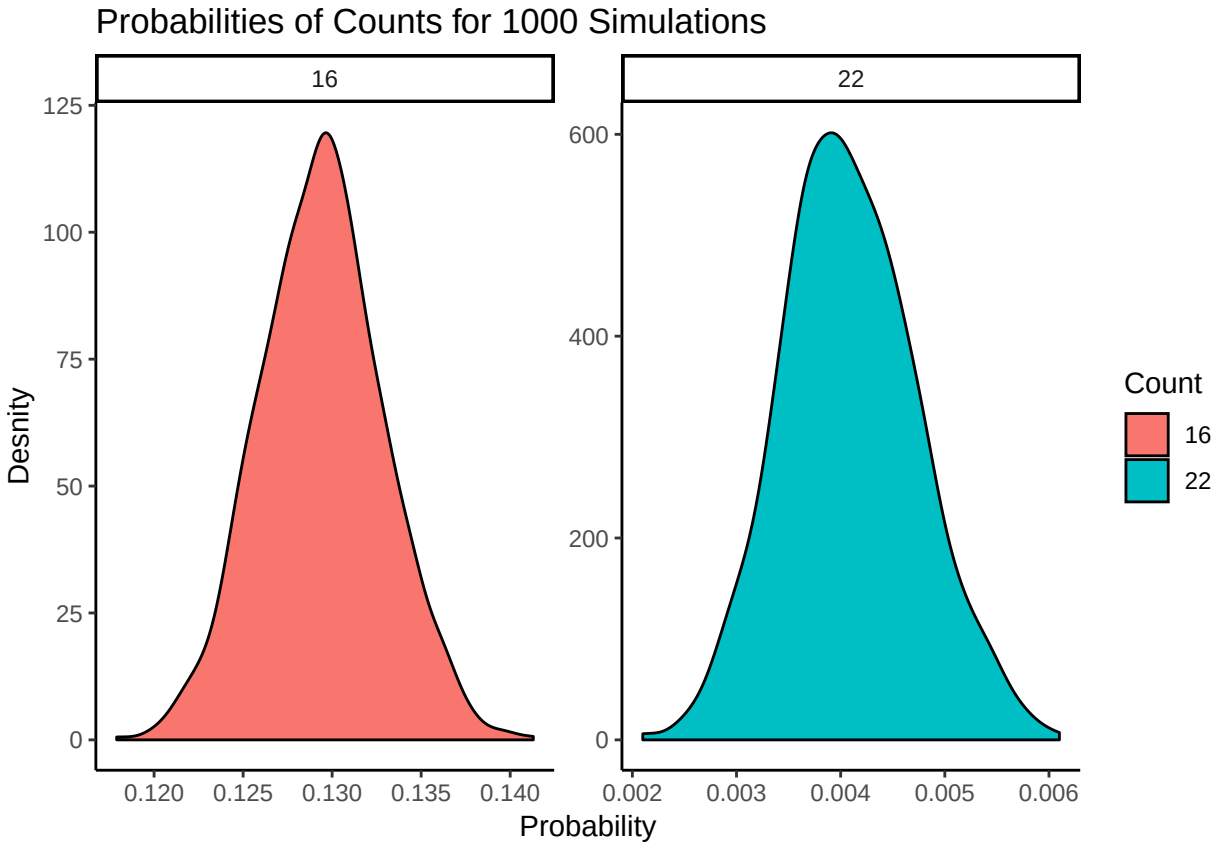
```
n <- 10000
singleDF <- tibble(out=rbinom(n=n, size=30, prob=.49), sim=1:n) %>%
    mutate(`Pr(x=22)`=cummean(out == 22), `Pr(x=16)`=cummean(out == 16)) %>%
    gather(key="Statistic", value="Probability", -out, -sim)

singleDF %>%
    ggplot(aes(x=sim, y=Probability, group=Statistic, color=Statistic)) +
    geom_line() +
    theme_classic() +
    geom_hline(yintercept=dbinom(22, size=30, prob=.49), linetype=3) +
    geom_hline(yintercept=dbinom(16, size=30, prob=.49), linetype=3) +
    labs(x="Simulation") +
    ggtitle("Cumulative Estimate of Probability of Outcome")
```



Notice that for the higher probability result we have more variation in the estimate across simulations which is a consequence of the binomial distributions variance. We can also estimate the true mean by taking n simulations where we take m draws from the binomial distribution of k trails and calculate the proportion of values that are equal to either 16 or 22. In the end we are left with a set of estimates that are approximately normally distributed around the true value.

```
m <- 1000 # Number of simulations (for plotting distribution)
n <- 10000 # number of replications (for calculating individual probs)
tibble(
    Probability=t(sapply(1:m, function(x){
        # Replicate binomial distribution n times with size k and prob .49
        sims <- rbinom(n=n, size=30, prob=.49)
        # calculate probabilty of being
        c(sum(sims == 22) / n, sum(sims == 16) / n)})) %>%
        c,
    Count=rep(c("22", "16"), each=m)) %>%
    ggplot(aes(x=Probability, fill=Count)) +
    geom_density() +
    facet_wrap(~Count, scales="free") +
    theme_classic() +
    labs(y="Desnity", title="Probabilities of Counts for 1000 Simulations")
```



Question 2

Consider two random variables X and Y. X can take values equal to 1, 2, or 3, and Y can take values equal to 1, 2, 3, 4. We know the following joint probabilities:

$f_{X,Y}(1,1) = .05$ $f_{X,Y}(2,1) = .07$ $f_{X,Y}(3,1) = .06$
$f_{X,Y}(1,2) = .08$ $f_{X,Y}(2,2) = ???$ $f_{X,Y}(3,2) = .07$
$f_{X,Y}(1,3) = .13$ $f_{X,Y}(2,3) = .16$ $f_{X,Y}(3,3) = .15$
$f_{X,Y}(1,4) = .03$ $f_{X,Y}(2,4) = .04$ $f_{X,Y}(3,4) = .06$

Using what you know about the relationships between marginal, conditional, and joint probability functions and the information above, compute the following quantities by hand (you should check your results in `R`):

We will place the results of the probabilities presented in a matrix. The last missing cell can be calculated by taking the some of the other probabilities and subtracting that value from 1.

```
probs <- c(.05, .08, .13, .03, .07, NA, .16, .04, .06, .07, .15, .06) %>%
    matrix(4, 3, dimnames=list(paste0("y=", 1:4), paste0("x=", 1:3)))

probs["y=2","x=2"] <- 1 - sum(probs, na.rm=T)

probs
```

```
##      x=1  x=2  x=3
## y=1 0.05 0.07 0.06
## y=2 0.08 0.10 0.07
## y=3 0.13 0.16 0.15
## y=4 0.03 0.04 0.06
```

Calculations by hand are done by taking the appropriate cells that relate to certain outcomes but we may also use R in order to easily calculate marginal probabilities by taking row sums and column sums. A check at the end shows us that all the results are the same.

```
## Hand calculations
(hcalc <- c(
    a=.05 + .08 + .13 + .03,
    b=.07 + .10 + .16 + .04,
    c=.06 + .07 + .15 + .06,
    d=.05 + .07 + .06,
    e=.08 + .10 + .07,
    f=.13 + .16 + .15,
    g=.03 + .04 + .06,
    h=.1,
    i=.08 / sum(.08 + .10 + .07),
    j=.06 / sum(.03 + .04 + .06),
    k=.06 / sum(.06 + .07 + .15 + .06),
    l=.10 / sum(.07 + .10 + .16 + .04)
))
```

```
##         a         b         c         d         e         f         g
## 0.2900000 0.3700000 0.3400000 0.1800000 0.2500000 0.4400000 0.1300000
##         h         i         j         k         l
## 0.1000000 0.3200000 0.4615385 0.1764706 0.2702703
```

```
## R calculations
(rcalc <- c(
    a=sum(probs[,"x=1"]),
    b=sum(probs[,"x=2"]),
    c=sum(probs[,"x=3"]),
    d=sum(probs["y=1",]),
    e=sum(probs["y=2",]),
    f=sum(probs["y=3",]),
    g=sum(probs["y=4",]),
    h=.1,
    i=probs["y=2","x=1"] / sum(probs["y=2",]),
    j=probs["y=4","x=3"] / sum(probs["y=4",]),
    k=probs["y=4","x=3"] / sum(probs[,"x=3"]),
    l=probs["y=2","x=2"] / sum(probs[,"x=2"])
))
```

```
##         a         b         c         d         e         f         g
## 0.2900000 0.3700000 0.3400000 0.1800000 0.2500000 0.4400000 0.1300000
##         h         i         j         k         l
## 0.1000000 0.3200000 0.4615385 0.1764706 0.2702703
```

```
cat(all.equal(hcalc, rcalc))
```

```
## TRUE
```