# HW1

Connor Hanna

2022-08-31

## Data

Code for the data object used in problems 1.4, 1.15 & 1.17

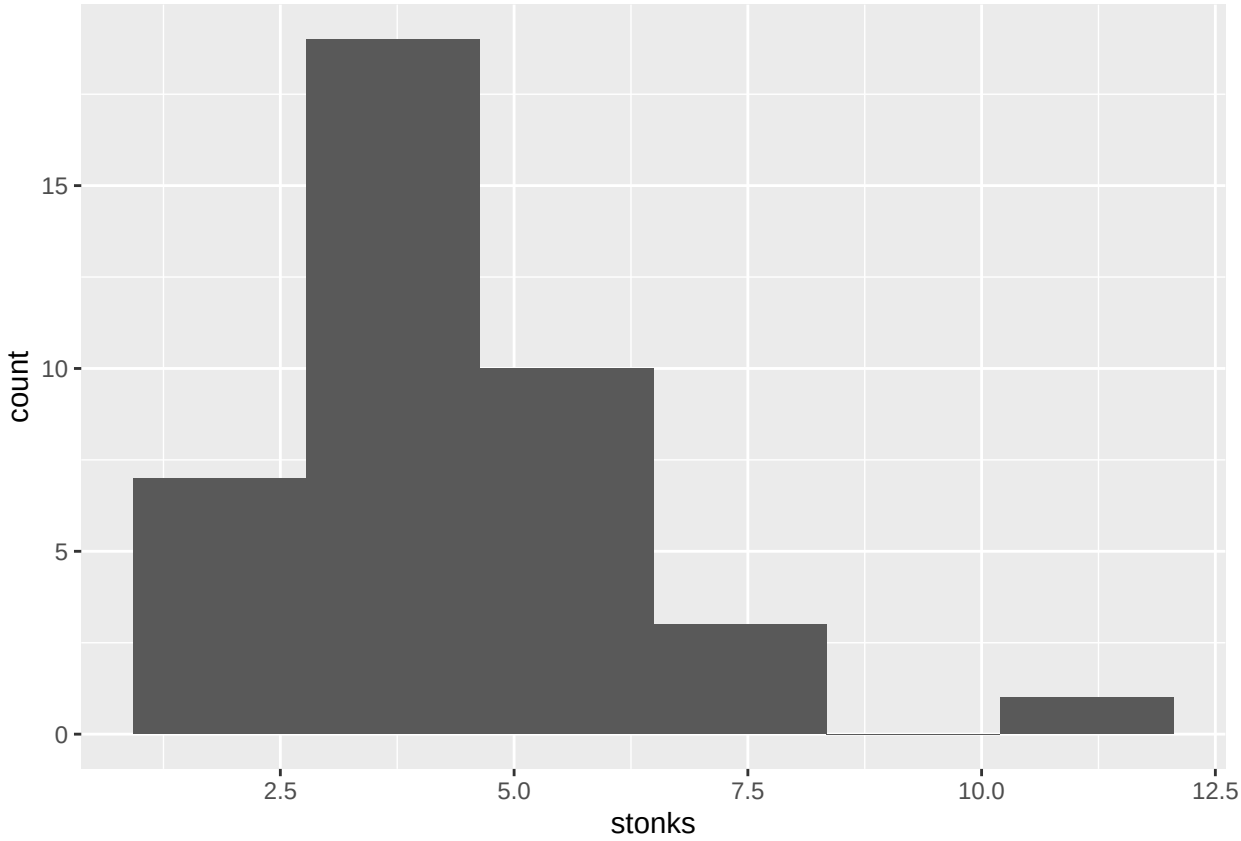
```
stonks <-
  c(11.88, 7.99, 7.15, 7.13, 6.27, 6.07, 5.98, 5.91, 5.49, 5.26, 5.07, 4.94,4.81, 4.79,
    4.55, 4.43, 4.40, 4.05, 3.94, 3.93, 3.78, 3.69, 3.62, 3.48, 3.44, 3.36, 3.26, 3.20,
    3.11, 3.03, 2.99, 2.89, 2.88, 2.74, 2.74, 2.69, 2.68, 2.63, 2.62, 2.61)

stonksdf <- data.frame(stonks)
```

## 1.4

### a

```
ggplot(data = stonksdf, aes(stonks)) +
  geom_histogram(bins = 6)
```




**b**

```
sum(stonksdf$stonks > 4)/count(stonksdf)
```

```
##      n
## 1 0.45
```

So the proportion of stocks traded at above 4% were 0.45.

**c**

```
sum(stonksdf$stonks < 5)/count(stonksdf)
```

```
##       n
## 1 0.725
```

So the proportion of stocks traded at below 5% were 0.725.

## 1.15

### a

```
mstonk <- mean(stonks)

vstonk <- var(stonks)

sdstonk <- sd(stonks)

mstonk
```

```
## [1] 4.387
```

```
vstonk
```

```
## [1] 3.502078
```

```
sdstonk
```

```
## [1] 1.871384
```

So the sample mean is 4.387, the sample variance is 3.502, and the sample standard deviation is 1.871.

### b

```
kstonk <-
  function(x){
    lower <- (mstonk - x * sdstonk)
    upper <- (mstonk + x * sdstonk)
    c(upper, lower)
  }

kstonk(1)
```

```
## [1] 6.258384 2.515616
```

```
kstonk(2)
```

```
## [1] 8.1297679 0.6442321
```

```
kstonk(3)
```

```
## [1] 10.001152 -1.227152
```

```r
nstonk <-
  function(x){
    lower <- I(mstonk - x * sdstonk)
    upper <- I(mstonk + x * sdstonk)
    stonks_df <- data.frame(stonks)
    vals <- filter(stonks_df, stonks > lower, stonks < upper)
    count(vals)
  }

nstonk(1)
```

```
##    n
## 1 35
```

```r
nstonk(2)
```

```
##    n
## 1 39
```

```r
nstonk(3)
```

```
##    n
## 1 39
```

So the bounds for the intervals are 6.258384-2.515616, 8.1297679-0.6442321, and 10.001152-(-1.227152) respectively for k = 1, 2, 3. The counts are 35, 39, 39, and in percentage terms the intervals contain 87.5%, 97.5%, and 97.5% of entries respectively. This is notably higher than the estimates for the empirical rule, likely because of the relative influence of outliers on the upper bound and also because the distribution has a distinct right skew.

## 1.17

```r
range <- max(stonks) - min(stonks)
range
```

```
## [1] 9.27
```

```r
appx <- range/4
appx
```

```
## [1] 2.3175
```

The range is 9.27 and the approximate standard deviation via the empirical rule is 2.317. This is higher than the calculated standard deviation, likely because of the skew in the distribution.

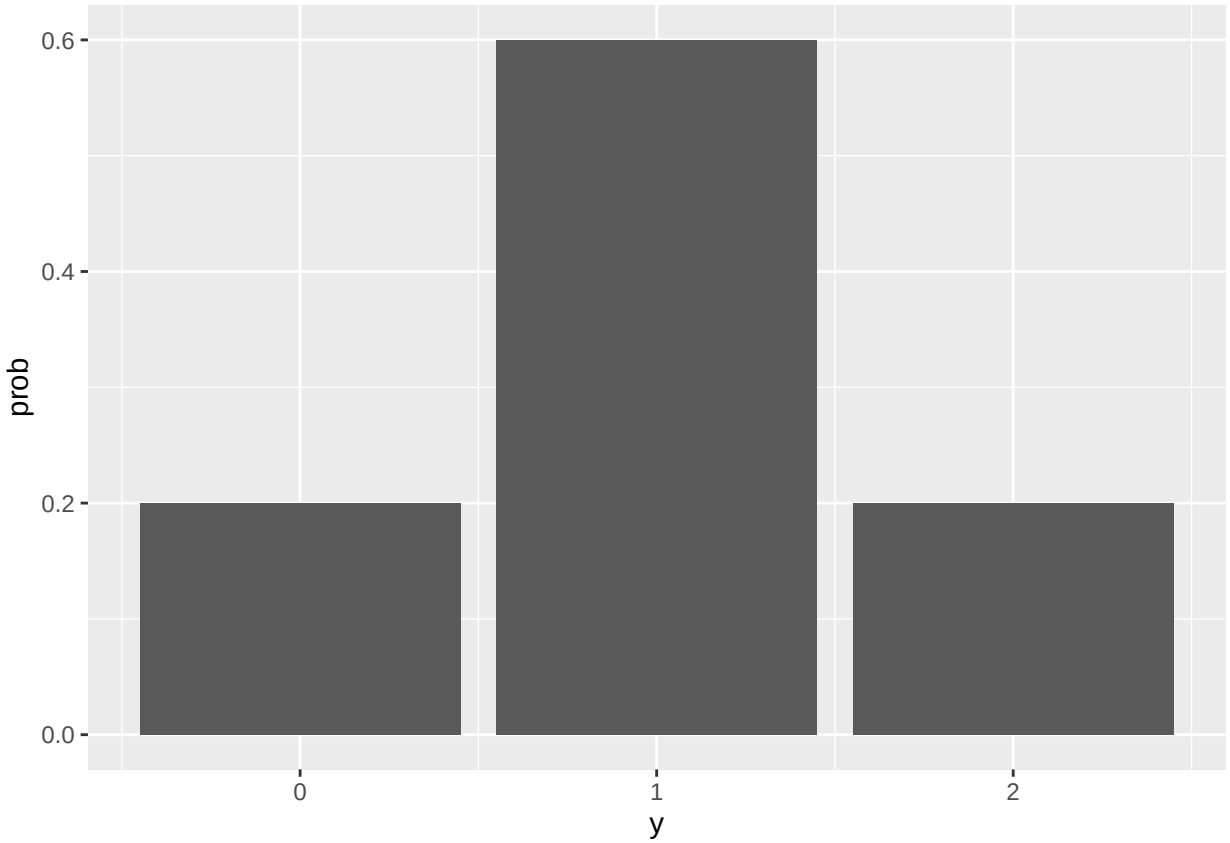
## 3.115

```
dhyper(0:2, 2, 4, 3)
```

```
## [1] 0.2 0.6 0.2
```

```
pop <- data.frame(prob = dhyper(0:2, 2, 4, 3), y = 0:2)

ggplot(data = pop, aes(y, prob)) +
  geom_col()
```

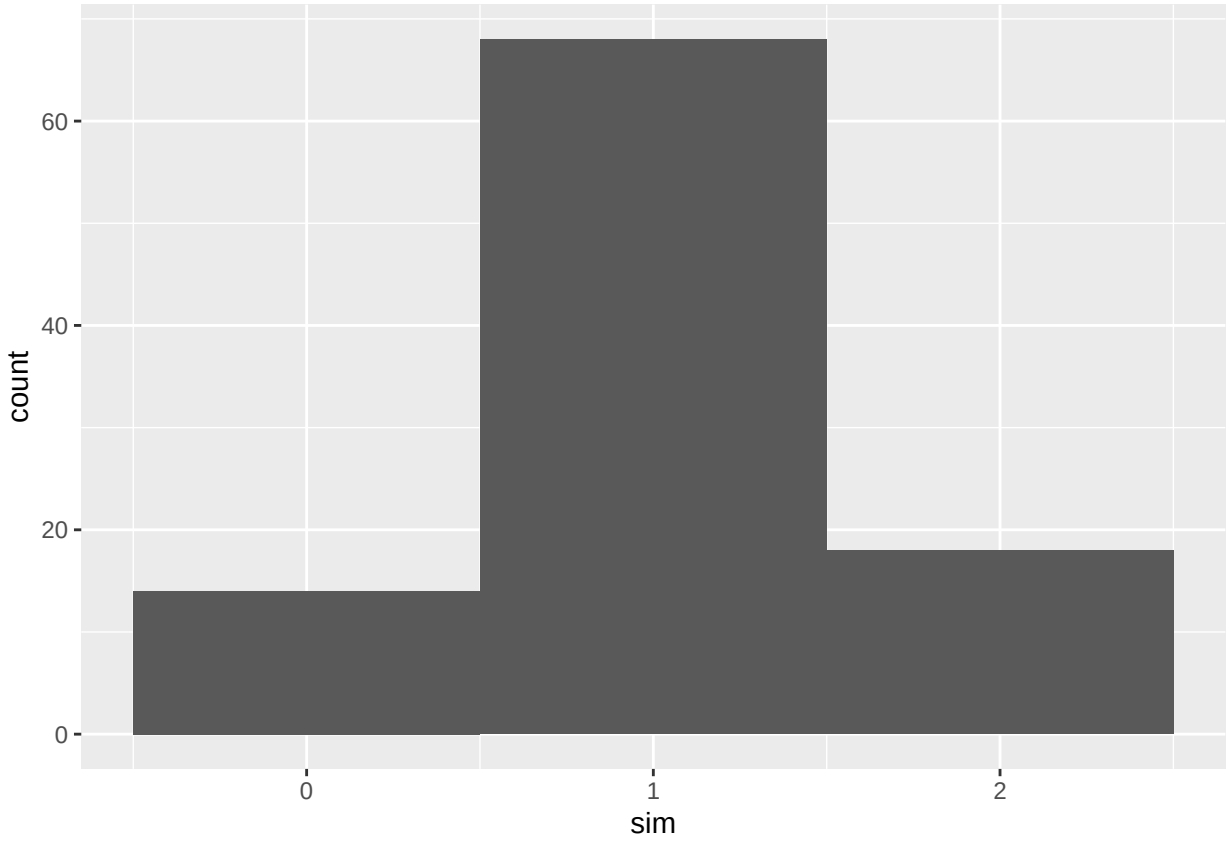



Per the output, the probabilities for P(y = 0), P(y = 1), and P(y = 2) are 0.2, 0.6, and 0.2 respectively.

### 3.116

```
sim <- data.frame(sim = rhyper(100, 2, 4, 3))

sim %>%
ggplot(aes(sim)) +
  geom_histogram(bins = 3)
```




Simulation of experimental data generation and the accompanying histogram generated above.

### 3.190

```
pop_mean <- sum(pop$prob * pop$y)
pop_mean
```

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

```
sample_mean <- mean(sim$sim)
sample_mean
```

```
## [1] 1.04
```

The mean based on the probability distribution/population is 1, and the sample mean is 1.06. The sample mean does provide a good estimate of the population mean.

### 3.191

```r
p <- .25
pop_var <- 3*p*(1-p)*(5/7)
#the above are formulas from the dhyper documentation
pop_var
```

```
## [1] 0.4017857
```

```r
sample_var <- var(sim$sim)
sample_var
```

```
## [1] 0.3216162
```

The population variance is 0.4017857, and the sample variance is 0.3939394. The sample variance is an excellent approximation of the population variance.

## 4.18

See additional hand work in accompanying PDF.

## 4.22

See additional hand work in accompanying PDF.