Some simulations

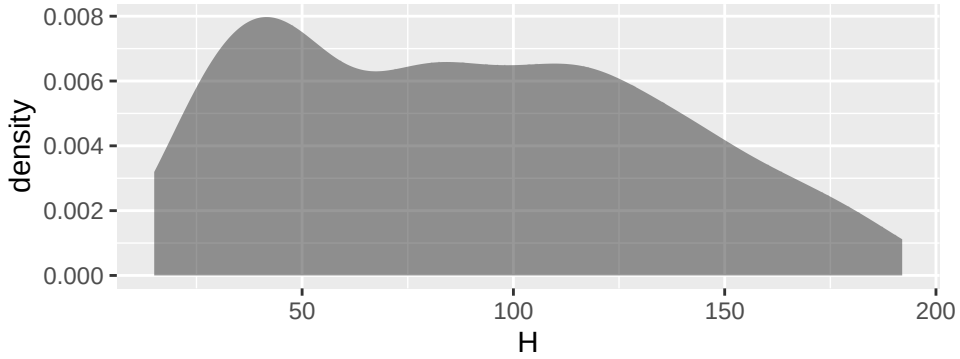
T.Scofield

You may click here to access the .qmd file.

Sample distributions take on the shape of the population distribution

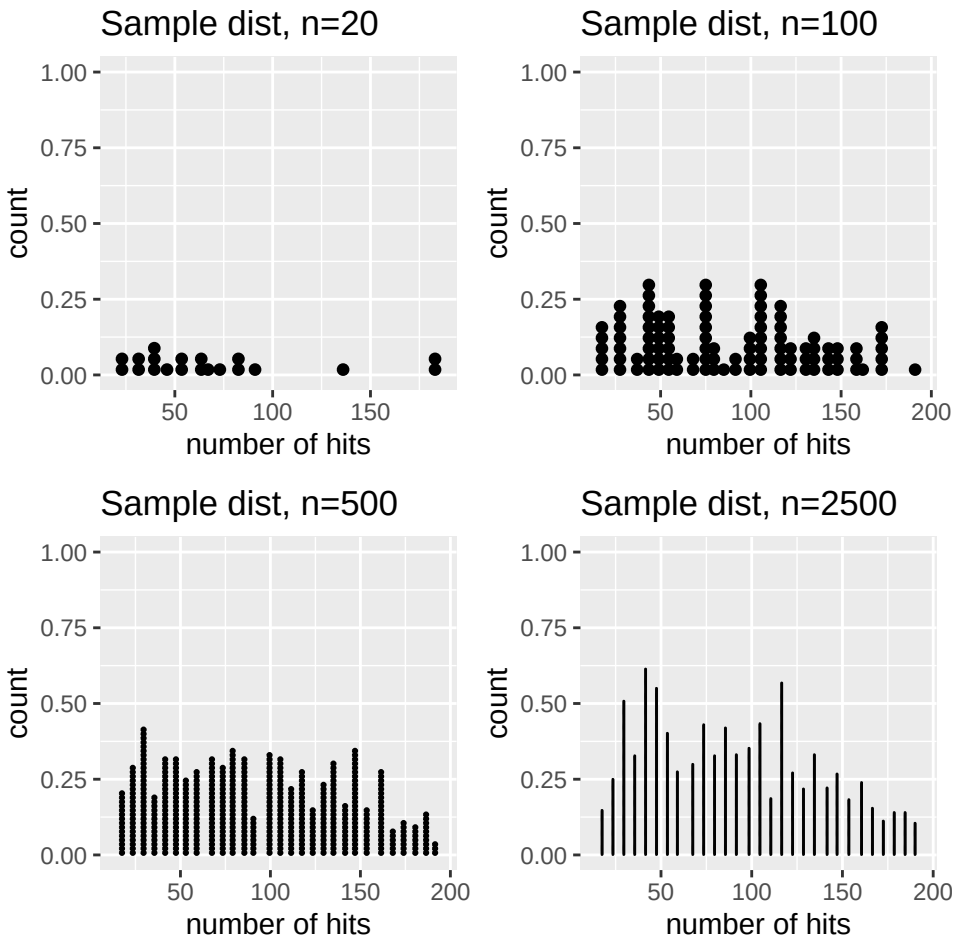
If I am sampling from `hits` (`H`) in the MLB18 data:

```
mlb18hitting <- read.csv("https://scofield.site/teaching/data/csv/mlb18abEligible.csv")
gf_density(~H, data=mlb18hitting)
```



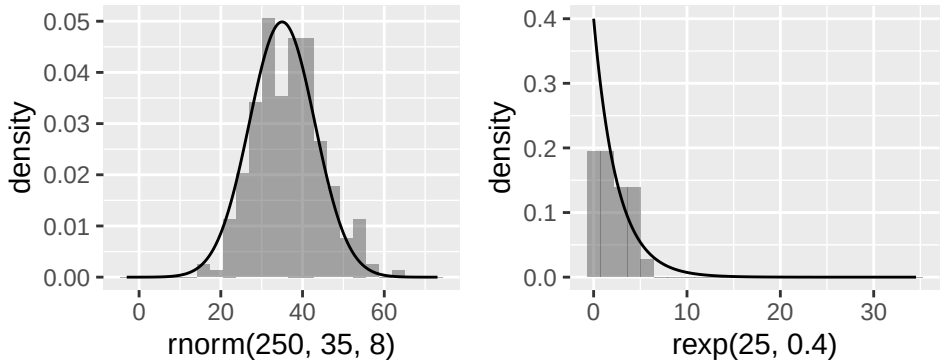
As we grow the sample size, the distribution of the sample takes on the shape of the population (given above):

```
p1 <- gf_dotplot(~resample(mlb18hitting$H, size=20)) |>
  gf_labs(title="Sample dist, n=20", x="number of hits")
p2 <- gf_dotplot(~resample(mlb18hitting$H, size=100)) |>
  gf_labs(title="Sample dist, n=100", x="number of hits")
p3 <- gf_dotplot(~resample(mlb18hitting$H, size=500), dotsize=.4) |>
  gf_labs(title="Sample dist, n=500", x="number of hits")
p4 <- gf_dotplot(~resample(mlb18hitting$H, size=2500), dotsize=.1) |>
  gf_labs(title="Sample dist, n=2500", x="number of hits")
grid.arrange(p1, p2, p3, p4)
```



The r-prefix commands draw samples from specified members of a distributional family. Again, if sample sizes are reasonably large, the underlying shape of the population begins to emerge. On the left, I have drawn a sample of size 250 from a Norm$(35, 8)$ distribution, then combined the two into an overlay. On the right, I have drawn a sample of size 25 from an Exp(0.4) distribution.

```
p1 <- gf_dhistogram(~rnorm(250, 35, 8)) |>
  gf_dist("norm", params=c(35,8))
p2 <- gf_dhistogram(~rexp(25, 0.4)) |>
  gf_dist("exp", params=c(0.4))
grid.arrange(p1, p2, ncol=2)
```



Simulating probabilities

A straight flush

In my "deck", the number 0 represents the Ace of Clubs, 1 represents the Two of Clubs, 2 represents the Three of Clubs, … 12 represents the King of Clubs, 13 represents the Ace of Diamonds, … 25 represents the King of Diamonds, … 39 represents the Ace of Spades, … and 51 represents the King of Spades.

This cell deals cards, then assesses if the cards are

- from the same suit, and
- in an unbroken sequence

```
deck = 0:51    # the deck
hand = sample(deck, size=5); hand    # draws 5 cards
```

```
[1] 29 21 12 51 25
```

```
suits = trunc( hand/13 ); suits     # suits of the hand
```

```
[1] 2 1 0 3 1
```

```
max(hand)-min(hand) == 4 & max(suits)-min(suits) == 0
```

```
[1] FALSE
```

The output is TRUE or FALSE, depending on whether the hand was a straight flush. We simulate many hands, and the long-term relative frequency should approximate the probability of a straight flush. A million runs may take a while, but rare events may not occur at all without a large number of iterations.

```
boolFlushResults <- replicate( 1000000, {
  hand = sample(deck, size=5);
  suits = trunc( hand/13 );
  max(hand)-min(hand) == 4 & max(suits)-min(suits) == 0
 }
)
sum(boolFlushResults) / 1000000
```

```
[1] 1.3e-05
```

Simulating a hypergeometric probability

The table on p. 99 gives rise to a data frame:

```
twins <- rbind(
  do(2) * c("dizygotic", "convicted"),
  do(15) * c("dizygotic", "notConvicted"),
  do(10) * c("monozygotic", "convicted"),
  do(3) * c("monozygotic", "notConvicted")
)
colnames(twins) = c("type", "record")
head(twins)
```

```
       type       record
1 dizygotic    convicted
2 dizygotic    convicted
3 dizygotic notConvicted
4 dizygotic notConvicted
5 dizygotic notConvicted
6 dizygotic notConvicted
```

We use this raw data to reconstruct the table:

```
tally(type ~ record, data=twins)
```

```
             record
type          convicted notConvicted
  dizygotic           2           15
  monozygotic        10            3
```

The number convicted among monozygotic twins is 10, and accessed from the array/table as

```
tally(type ~ record, data=twins)[2,1]
```

```
[1] 10
```

It seems somewhat on the high side, particularly when you view there are only 13 monozygotic twins in the study, and the number of convicted dizygotic twins is a small proportion of the 17 of those in the study.

But what hypotheses are we testing? It's

$$\mathbf{H}_0 : \text{'Type of twin' and 'whether convicted' are independent.}$$

It seems the alternative one would set out to prove is one-sided, reflecting a higher conviction rate among monozygotic twins. In that regard, our P-value should reflect results as extreme or more so than 10 in the bottom left cell. That is the value produced by the command

```
1-phyper(9, 13, 17, 12)
```

```
[1] 0.0004651809
```

We might simulate this probability using **permutation testing**. We maintain the marginal totals, but shuffle one variable. To do so, changes the counts in the table itself. Here is one run.

```
tally(type ~ shuffle(record), data=twins)
```

```
             shuffle(record)
type          convicted notConvicted
  dizygotic           6           11
  monozygotic         6            7
```

When we repeat this many times, we will grab just the value in the 2nd row, 1st column.

```
simCornerValue <- replicate(50000,
          tally(type ~ shuffle(record), data=twins)[2,1]
          )
head(simCornerValue)
```

```
[1] 6 7 6 4 5 6
```

We find the proportion of values higher than 10:

```
prop(~(simCornerValue >= 10))
```

```
prop_TRUE
  0.00046
```

This is reasonably close to the value 0.0004652 found earlier using `phyper()`.