R Tutorial-12

T. Scofield

You may click here to access the .qmd file.

In this issue, we

- manipulate the display of factors
- build tables (summaries which provide the distributions of factors)
- carry out inference on a single proportion

Factors

The term **factor** is used in the R community for a variable with finitely many values, particularly if those values split the data into separate categories—think `class standing` for students, `make` for automobiles, or `number of axles` for assessing vehicle tolls on the highway. As the last of these shows, a factor may be quantitative, yet used like a categorical variable.

Sometimes data frames contain variables which are factors, yet have codified the values as numbers—an example might be a variable `sex` whose values are 0 and 1. The `wrist` data frame in the **fosdata** package has variables `handed_side` and `fracture_side`, meant to indicate which hand is a patient's dominant hand, and which hand the patient fractured. Since the data frame (or **tibble**) has 47 columns/variables, we select out just these two:

```
dim(wrist)
```

```
[1] 105  47
```

```
wrist |> select(handed_side, fracture_side) |> head()
```

```
# A tibble: 6 x 2
  handed_side fracture_side
        <dbl>         <dbl>
1           1             2
2           1             1
3           1             1
4           1             2
5           1             1
6           1             2
```

After running `help(wrist)`, we learn that a 1 is used for right hand, 2 for left. We can alter our data to make this more explicit via

```
wristFeaturedVars = wrist |> select(handed_side, fracture_side) |>
  mutate(
  handed_side = factor(handed_side, labels=c("right","left")),
  fracture_side = factor(fracture_side, labels=c("right","left"))
)
head(wristFeaturedVars)
```

```
# A tibble: 6 x 2
  handed_side fracture_side
  <fct>       <fct>
1 right       left
2 right       right
3 right       right
4 right       left
5 right       right
6 right       left
```

Tables

There are multiple commands (some made available by add-on packages) that can build frequency tables in R. I have eschewed the use of `table()` (the most obvious, and used regularly in the text), because it does not accept the formula notation `varY` ~ `varX` used with **mosaic** commands. Instead, I have often used `tally()` (**mosaic**). In what follows, I will use `xtabs()`. I suggest you try the next two commands both as written, and with `tally()` in place of `xtabs()`, to gain an appreciation for the change.

```
xtabs(~fracture_side, data=wristFeaturedVars)
```

```
fracture_side
right  left
   45    60
```

```
xtabs(~ fracture_side + handed_side, data=wristFeaturedVars)
```

```
             handed_side
fracture_side right left
        right    41    3
        left     56    4
```

Running these commands, but drawing data from the original `wrist` data frame (where factor levels are still given as 1 and 2) makes it clear what advantage is afforded by the `mutate()` command above.

```
xtabs(~fracture_side, data=wrist)
xtabs(~ fracture_side + handed_side, data=wrist)    # I have blocked evaluation of these
```

One also may appreciate, particularly for the two-way table above, seeing row and column totals, achieved by piping the table to `addmargins()`

```
xtabs(~ fracture_side + handed_side, data=wristFeaturedVars) |> addmargins()
```

```
             handed_side
fracture_side right left Sum
        right    41    3  44
        left     56    4  60
        Sum      97    7 104
```

or obtaining relative frequencies (proportions) in each cell instead of frequencies

```
handTable = xtabs(~ fracture_side + handed_side, data=wristFeaturedVars)
handTable |> proportions()
```

```
             handed_side
fracture_side      right       left
        right 0.39423077 0.02884615
        left  0.53846154 0.03846154
```

The code in this last chunk demonstrates that

- `xtabs()` employs raw data to build a summary, the distribution of factors
- the resulting summary, a table, can be stored and used for further processing

Note that `proportions()`, by default, produced proportions out of the total of 104 hand-fracture patients; that is, adding up all four proportions equals 1. Try out (on your own, as I have suppressed evaluation of these commands) these variants:

```
handTable |> proportions(margin=1)
handTable |> proportions(margin=2)
```

Building a table directly from the numbers for its cells

Many books, newspapers and the like display their data already summarized in table form. You may find you wish to build an R table directly containing the frequencies you see from an article. The website https://www.statology.org/two-way-table-in-r/ displays the following table:

	Baseball	Basketball	Football	Total
Male	13	15	20	48
Female	23	16	13	52
Total	36	31	33	100

The Statology site offers instructions, based on the `matrix()` command, for building this table. After some modification, I give those instructions here. Keep in mind that we are superceding the work of the `xtabs()` command and, in fact, we have not acquired an actual data set at all, which is what `xtabs()` would require.

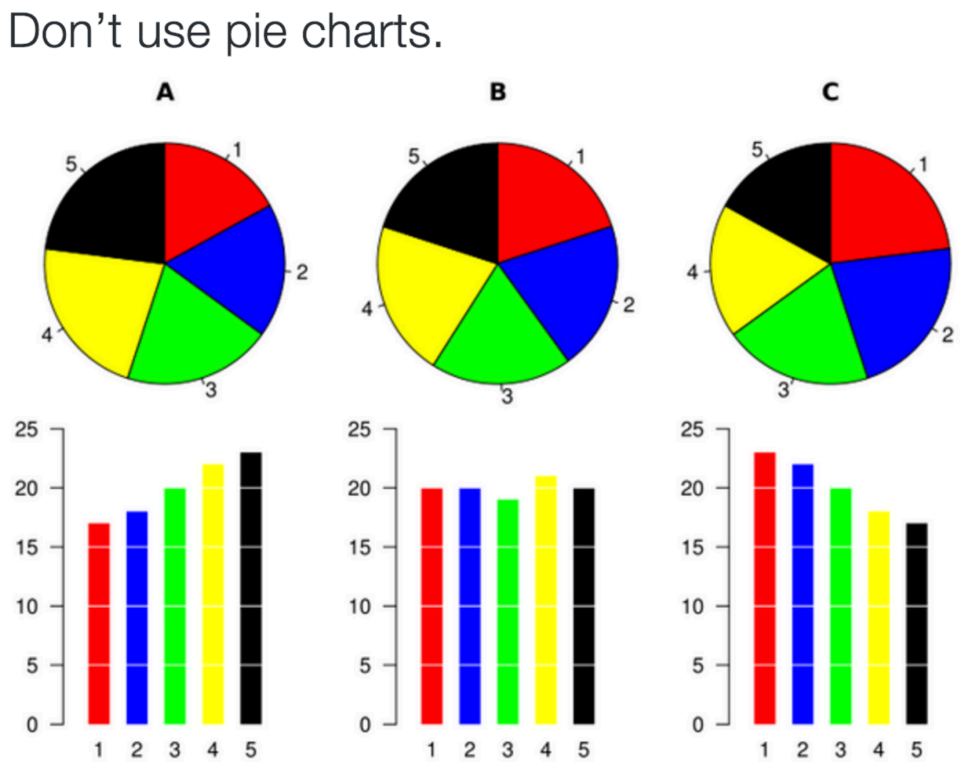
```
myTable <- matrix(c(13, 15, 20, 23, 16, 13), ncol=3, byrow=TRUE)
rownames(myTable) = c("Male", "Female")
colnames(myTable) = c("Baseball", "Basketball", "Football")
myTable
```

```
       Baseball Basketball Football
Male         13         15       20
Female       23         16       13
```

```
# myTable |> addmargins()   # Variant to throw in marginal totals
```

Visual displays

There are various ways to display factors/categorical data. One I tend to denigrate is a pie chart. Here is a graphic showing three different distributions of a categorical variable in both pie chart and bar graph form. There are subtle differences in the distributions for Data A, Data B and Data C, and the bar graphs do a better job eliciting these differences.



As for examples of generating bar graphs using `gf_bar()`, I recommend my R Tutorial Episode 9, beginning about half way through page 8 with the heading bar charts without and with slicing.

One-proportion inference

Any population proportion p represents a binary focus:

- "What proportion of American citizens are octogenarians?" focuses on American citizens and not so much their ages as whether that age is ≥ 80 and < 90 (yes/no).
- "What proportion of people have blood type A-positive?" glosses over that there are other blood types, reducing to a yes/no-answer (binary) about being A-pos.

Some facts about proportions

1. A sample proportion is a ratio X/n where n is sample size, and $X = X_1 + X_2 + \cdots + X_n$ where each X_i is 0 or 1, depending on the success or not of a Bernoulli trial. Said another way,

$$\hat{p} = \frac{1}{n} \sum_i X_i$$

   is the mean of a random sample.
2. If the X_i come as an iid sample, then the numerator $X \sim \text{Binom}(n, p)$, where p is the population proportion. In practice, the X_i are often not iid, but if n is relatively small in comparison to the size of the population, we can treat an SRS the same as iid. This enables us to test a null hypothesis

$$\mathbf{H}_0 \colon p = p_0$$

   against a 1- or 2-sided alternative using `binom.test()` much as we did for the **sign test**; the only modification is we will use p_0 (the proposed null value), in the function call (the sign test had us using 1/2 for the null value). The command also offers confidence intervals for p in its output.

Example 1: A college football team has a returning kicker who, last season, made only 40% of field goals at distances 50 yards or more. They offer a freshman walk-on the chance to win the job, giving this newbie $n = 10$ attempts at 50+ yards. The young man gets the position if his test statistic is significant at the 5% level, when his tryout is viewed as an hypothesis test of

$$\mathbf{H}_0 \colon p = 0.4 \qquad \text{vs.} \qquad \mathbf{H}_a \colon p > 0.4.$$

Suppose the player makes 6 of 10; here, $X = 6$ will serve as his test statistic. Using `binom.test()`, the relevant command is

```
binom.test(6, 10, 0.4, alternative="greater")
```

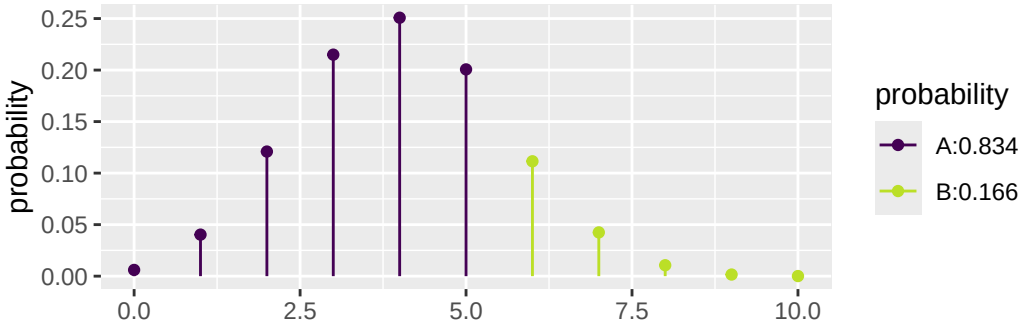
```
data:  6 out of 10
number of successes = 6, number of trials = 10, p-value = 0.1662
alternative hypothesis: true probability of success is greater than 0.4
95 percent confidence interval:
 0.3035372 1.0000000
```

```
sample estimates:
probability of success
                   0.6
```

The P-value is 0.1662, not statistically significant, which means the evidence is not strong enough (when $\alpha = 0.05$) to reject that this walk-on is successful only 40% of the time at this distance.

To see, visually, what is going on, look at the graph produced by the command

```
1 - xpbinom(5, 10, 0.4)
```




```
[1] 0.1662386
```

This shows the null distribution Binom$(10, 0.4)$, coloring results "5 or less" one way, and results "6 or more" another. In a world where this young man's rate of success at 50+ yards is truly 40%, you will see him make "6 or more" out of 10 about 16.6% of the time. Our P-value omits things in the left tail, because it was a 1-sided (right-tailed) alternative hypothesis. You should be able to see that the command

```
1 - pbinom(5, 10, 0.4)
```

is a direct way of obtaining the P-value.

If one wishes to have a 90% confidence interval for p, the proportion of kicks at 50-plus yards this newbie makes, `binom.test()` can be tweaked to report this, too. (The above command gave a 1-sided 95% confidence interval, by the way.) We obtain (evaluation has been suppressed) a (2-sided) 90% confidence interval using

```
binom.test(6, 10, 0.4)
```

I put 0.4 in for the null value, but confidence intervals are constructed without taking into account null values. So you should get the same 90% confidence interval even if you change 0.4 to 0.1, 0.5, or whatever in the command above.

Example 2: It is said that, in the United States, 35.7% of people are of blood type A-positive. Let's take as our research question: Does the proportion p of South Koreans with blood type A+ match that in the U.S.? That is, we wish to test

$$\mathbf{H}_0 : p = 0.357 \qquad \text{vs.} \qquad \mathbf{H}_a : p \neq 0.357.$$

Now you collect a random sample (probably something akin to an SRS). Suppose 63 out of 211 Koreans sampled have blood type A+. We use the (default) two-sided alternative in our call to `binom.test()`:

```
binom.test(63, 211, 0.357)
```

```
data:  63 out of 211
number of successes = 63, number of trials = 211, p-value = 0.08445
alternative hypothesis: true probability of success is not equal to 0.357
95 percent confidence interval:
 0.2376893 0.3652278
sample estimates:
probability of success
             0.2985782
```

The P-value, 0.08445, is not statistically significant at the 5% level, so we cannot reject the null hypothesis (at that level) that South Koreans also have blood type A+ at a rate of 35.7%. (Note that Wikipedia indicates 32% of South Koreans have A+ blood. Assuming Wikipedia is correct in its value of the population proportion p, then our sample led us into a Type II error.)

Important Note: The P-value reported by `binom.test()` is calculated as

```
referenceProbability = dbinom(63, 211, 0.357)
binomProbabilities = dbinom(0:211, 211, 0.357)
sum(binomProbabilities[binomProbabilities <= referenceProbability])
```

```
[1] 0.08445173
```

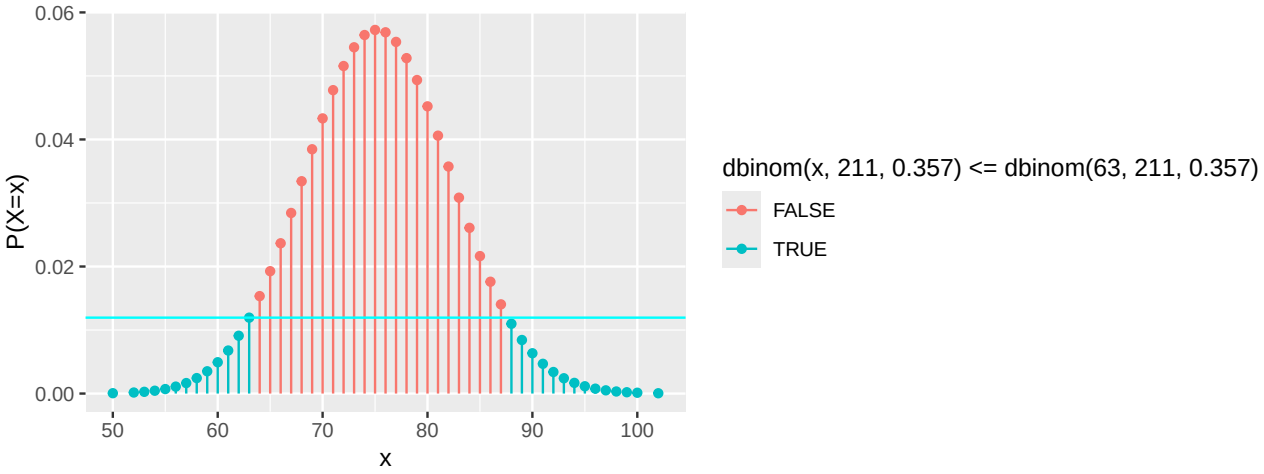
which is *not* the same as

```
2 * pbinom(63, 211, 0.357)
```

```
[1] 0.08648912
```

See the graphical depiction.

```
gf_dist("binom", size=211, prob=0.357,
        color = ~ dbinom(x,211,0.357) <= dbinom(63,211,0.357)) |>
  gf_labs(y="P(X=x)") |>
  gf_hline(yintercept=dbinom(63,211,.357), color="cyan")
```



The reason they differ is that the Binom$(211, 0.357)$ distribution is not truly symmetric, though it is not badly skewed. The more skewed your binomial (null) distribution, the greater this discrepancy will be.

3. Since $\hat{p}$ is a mean, the Central Limit Theorem says that it will have an approximately normal distribution when n is large enough. Given that

- the count of successes $X \sim \text{Binom}(n, p)$ has expected value np and variance $np(1-p)$, and
- $\hat{p} = \frac{X}{n}$,

it follows that $\hat{p}$ has an approximate sampling distribution $\text{Norm}\left(p, \sqrt{\frac{p(1-p)}{n}}\right)$, or equivalently that

$$\frac{\hat{p} - p}{\sqrt{\frac{p(1-p)}{n}}} \sim \text{Norm}(0, 1).$$

Using the normal approximation

Statistics books have, for many years, assumed people wanted to use a normal approximation for the sampling distribution of $\hat{p}$. This was largely because they could print a standard normal table, such as the one at https://math.arizona.edu/~jwatkins/normal-table.pdf (a substitute for `pnorm()`), and could not count on people having the computing power wrapped into `pbinom()`. They offered this rule of thumb, as a test for whether n is large enough to justify a normal approximation:

$$\text{One can treat } \hat{p} \sim \text{Norm}\left(p, \sqrt{\frac{p(1-p)}{n}}\right) \text{ so long as } np \geq 10 \text{ and } n(1-p) \geq 10.$$

Let's see this in action.

Example 3:

Once again, suppose there are 63 of 211 South Koreans with Type A+ blood. We wish, as before, to test hypotheses

$$\mathbf{H}_0 : p = 0.357 \qquad \text{vs.} \qquad \mathbf{H}_a : p \neq 0.357.$$

We note that, assuming the null hypothesis is true,

$$np = (211)(0.357) = 75.327 \qquad \text{and} \qquad n(1-p) = (211)(0.643) = 135.673,$$

so the rule-of-thumb for a normal approximation is justified. The reference distribution for $\hat{p}$ is

$$\text{Norm}\left(0.357, \sqrt{\frac{(0.357)(0.643)}{211}}\right) = \text{Norm}(0.357, 0.03298362).$$

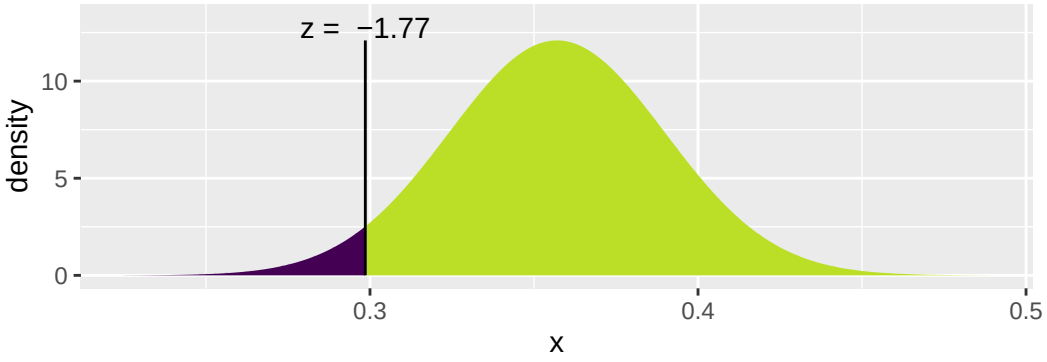
Taking our test statistic to be $\hat{p} = 63/211$ as our test statistic, the area in the right tail is

```
xpnorm(63/211, 0.357, 0.03298362)
```

```
If X ~ N(0.357, 0.03298), then

    P(X <= 0.2986) = P(Z <= -1.771) = 0.03826

    P(X >  0.2986) = P(Z >  -1.771) = 0.9617
```




```
[1] 0.03826067
```

As our alternative hypothesis is two-tailed, and normal distributions are always symmetric, we may take our P-value as twice this number:

```
2*pnorm(63/211, 0.357, 0.03298362)
```

```
[1] 0.07652134
```

It would be too strong to say the use of a normal approximation to the sampling distribution of $\hat{p}$ has fallen out of favor; it is still taught in most intro statistics courses. For those who use it, there is an obvious improvement, called **continuity correction** (named so because we are replacing a discrete distribution with a continuous one), which leads to selecting the numerator for $\hat{p}$ to be half-way between two integers. In our case, applying continuity correction yields the P-value

```r
2*pnorm(63.5/211, 0.357, 0.03298362)
```

```
[1] 0.08924516
```

prop.test()

The `prop.test()` command is as easy to use as `binom.test()`. It

- automates the use of a normal approximation, yet
- provides a warning alongside the results when the rule of thumb is not met, and
- does continuity correction by default.

```r
prop.test(63, 211, p=0.357)
```

```
    1-sample proportions test with continuity correction

data:  63 out of 211
X-squared = 2.8879, df = 1, p-value = 0.08925
alternative hypothesis: true p is not equal to 0.357
95 percent confidence interval:
 0.2386904 0.3659425
sample estimates:
        p
0.2985782
```

Like `binom.test()`, `prop.test()` produces a confidence interval (at the 95% level, by default). Since `prop.test()` is based on a normal approximation, its upper/lower bounds are not a match to those produced by `binom.test()`. The results of `binom.test()` (both confidence intervals and P-values) are more accurate than those of `prop.test()`.