Confidence intervals and hypothesis tests in R

Víctor Peña

Contents

Confidence intervals and hypothesis tests **1**

One normal mean 1

Two independent normal means 3

One proportion 6

Two proportions 7

χ^2-tests of independence for categorical variables 7

Pairwise comparisons of multiple normal means: Tukey HSD 9

Confidence intervals and hypothesis tests

I'm assuming we're all familiar with confidence intervals and hypothesis testing. If you need a refresher, there are many good resources online. I think the videos on Khan Academy[1] are quite good, but feel free to consult other sources (let me know if you find any other good and freely available sources, so that I can recommend them to other students in the future).

One normal mean

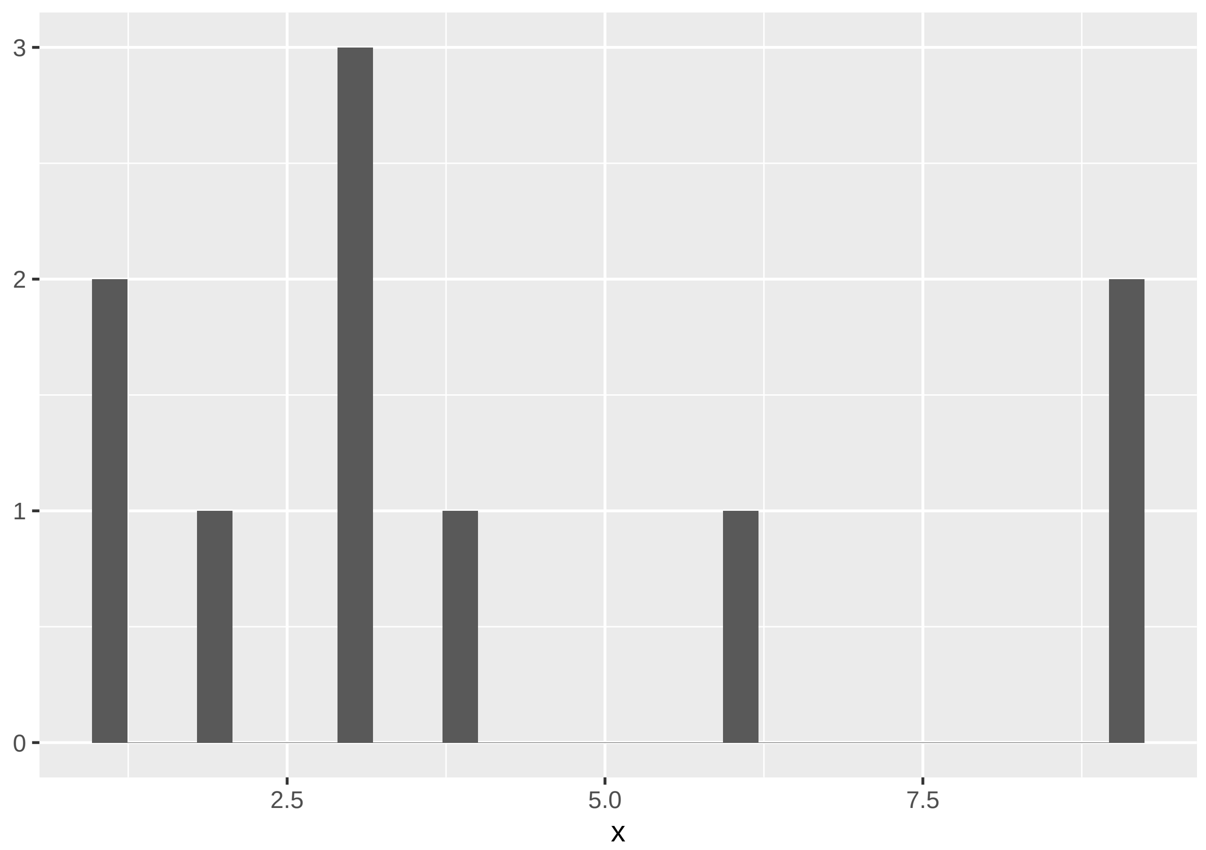
A group of scientists recorded some measurements that are stored in the vector `x`:

```r
x = c(1,4,2,3,6,9,1,3,9,3)
```

We can visualize the data using a histogram:

```r
library(tidyverse)
qplot(x)
```

[1] `https://www.khanacademy.org/math/statistics-probability`



We can do t-tests and find confidence intervals for the population mean μ with the function `t.test`.

For example, if we want to find a 99% confidence interval for the mean:

```
t.test(x, conf.level = 0.99)
```

```
##
##  One Sample t-test
##
## data:  x
## t = 4.3789, df = 9, p-value = 0.001774
## alternative hypothesis: true mean is not equal to 0
## 99 percent confidence interval:
##  1.057163 7.142837
## sample estimates:
## mean of x
##       4.1
```

We can change `conf.level` to any confidence level that we want. If we don't specify anything, the default is 95% confidence.

As you can see, `t.test` gives us a p-value. It is a p-value for the test $H_0 : \mu = 0$ against $H_1 : \mu \neq 0$. If we want to change the hypothesized value under the null to some value that is not 0, we can change the argument `mu` of `t.test`. If we want to change the alternative to "less than" or "greater than", instead of "not equal to", we can set `alternative` to `less` or `greater`, respectively.

For instance, if we want to test $H_0 : \mu = 5$ against $H_1 : \mu < 5$, the following `R` code will do it for us:

```
t.test(x, mu = 5, alternative = 'less')
```

```
##
##  One Sample t-test
##
## data:  x
## t = -0.96123, df = 9, p-value = 0.1808
## alternative hypothesis: true mean is less than 5
## 95 percent confidence interval:
##      -Inf 5.816352
## sample estimates:
## mean of x
##       4.1
```

As always, we can get more information about the function if we type in `?t.test`.

Two independent normal means

A pharmaceutical is interested in knowing whether their new treatment is significantly different than the current gold standard. They collected a sample of 40 individuals: 20 of them were assigned the new treatment, and 20 of them were assigned the current treatment. The outcome is on an ordinal scale that goes from 0 to 100, where 0 is "bad" and 100 is "great". We read in the data:

```
pharma = read.csv("https://vicpena.github.io/sta9750/fall18/pharma.csv")
```

The data has 2 columns: `group` and `outcome`:

```
summary(pharma)
```

```
##     group              outcome      
##  Length:40          Min.   : 80.00  
##  Class :character   1st Qu.: 89.00  
##  Mode  :character   Median : 92.50  
##                     Mean   : 92.30  
##                     3rd Qu.: 98.25  
##                     Max.   :100.00
```

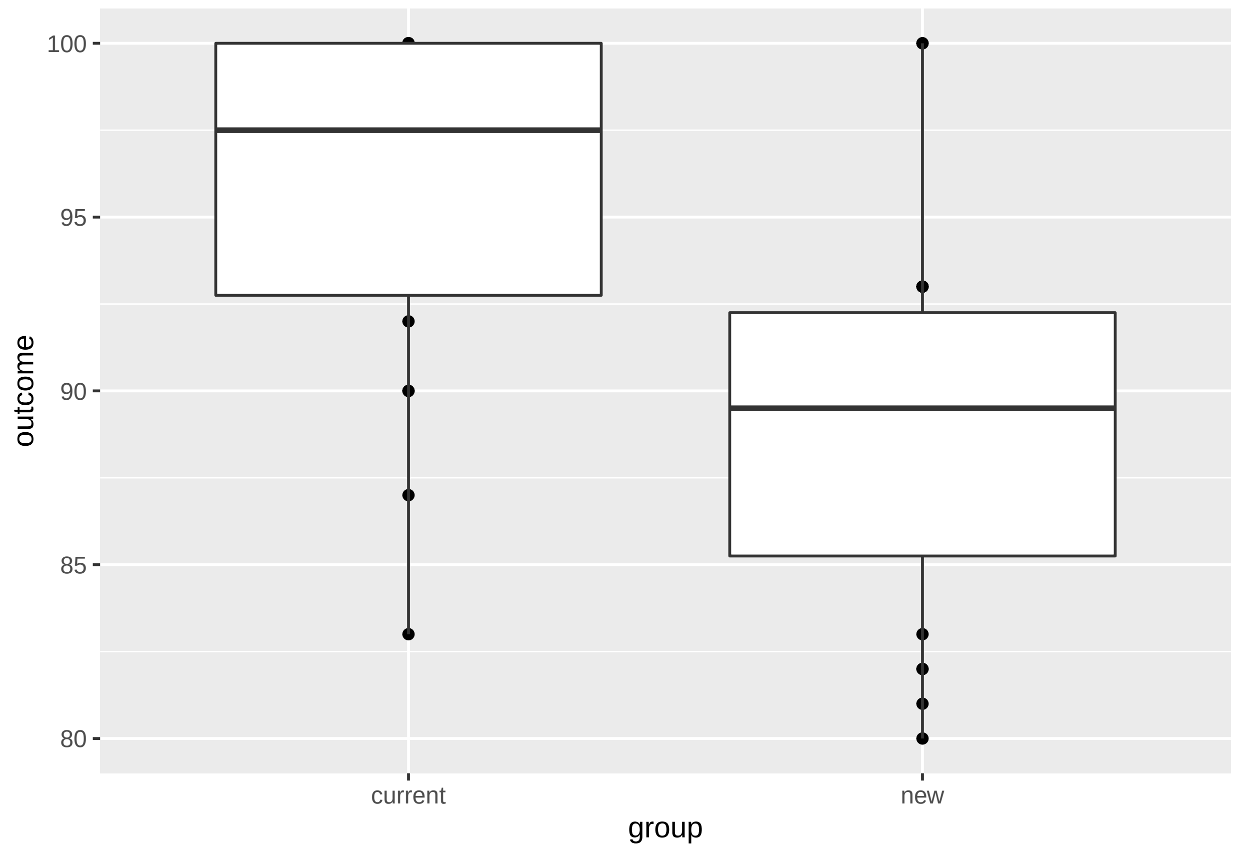
If we want to get summaries by group (means, standard deviations, etc.), there are different ways to do it. Here's one using `group_by` and `summarize`:

```
library(tidyverse)
pharma %>%
  group_by(group) %>%
    summarize(avg=mean(outcome), stdev= sd(outcome))
```

```
## # A tibble: 2 x 3
##   group     avg stdev
##   <chr>   <dbl> <dbl>
## 1 current  95.6  5.02
## 2 new      89.0  5.67
```

If we want to plot the outcomes by group, we can create a boxplot:

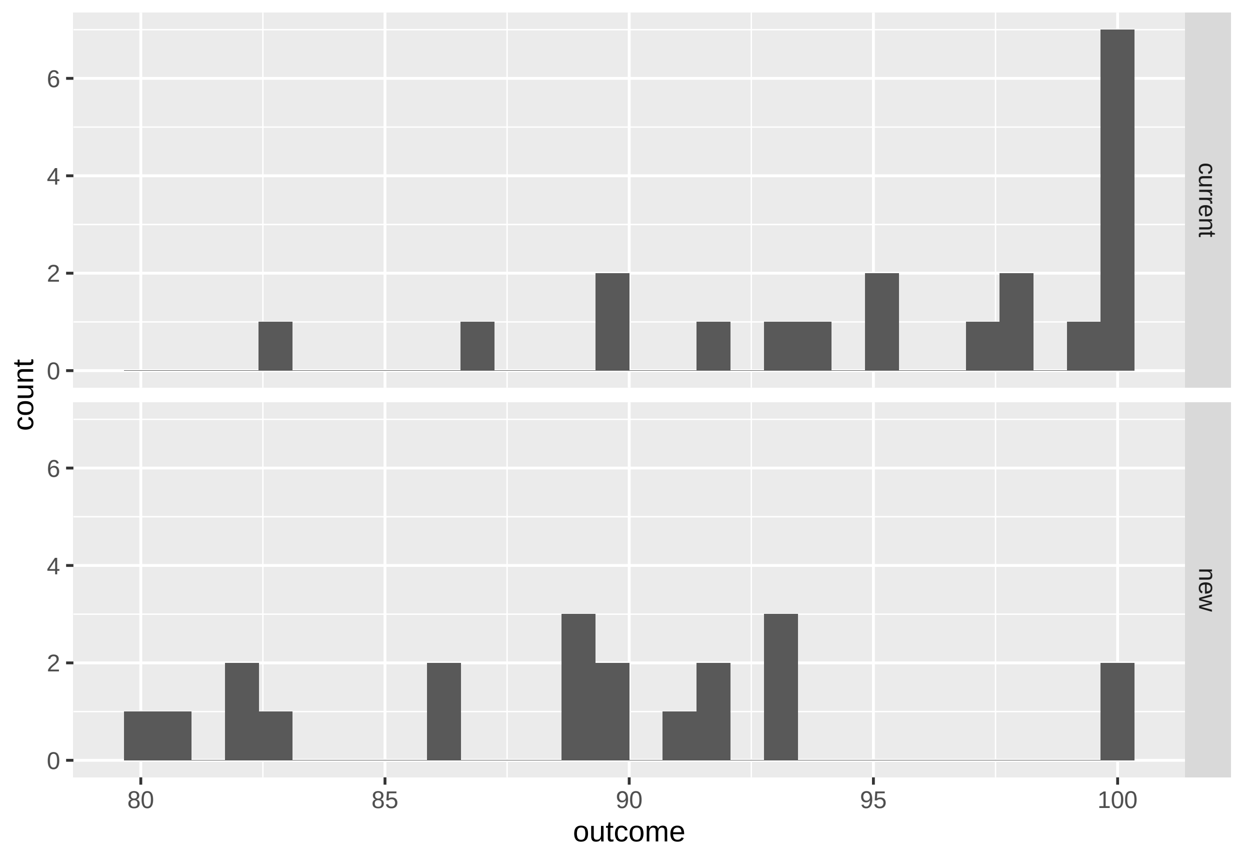
```r
qplot(y = outcome, x = group, data = pharma) + geom_boxplot()
```



If we want to create histograms of outcomes by groups, here's an option using `facet_grid`:

```r
ggplot(pharma) +
  aes(x = outcome) +
   geom_histogram() +
    facet_grid(group ~ .)
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



The pharmaceutical is interested in knowing whether the population means of the health outcomes are different at the 0.01 significance level. That is, if μ_C and μ_N are the population means of `outcome` for the current and new treatments, respectively, the pharmaceutical wants to test $H_0 : \mu_C = \mu_N$ against $H_1 : \mu_C \neq \mu_N$ at the 0.01 significance level.

We can do a two-sample t-test with `t.test`:

```
t.test(outcome ~ group, conf.level = 0.99, data = pharma)
```

```
##
##  Welch Two Sample t-test
##
## data:  outcome by group
## t = 3.8381, df = 37.451, p-value = 0.0004622
## alternative hypothesis: true difference in means is not equal to 0
## 99 percent confidence interval:
##   1.904267 11.095733
## sample estimates:
## mean in group current     mean in group new
##                 95.55                 89.05
```

We can change the hypothesized difference in means under the null with the argument `mu` and we can change the alternative to "less than" or "greater than" with the argument `alternative`, just as we did with the one-sample t-test we covered in the previous section.

The treatments are significantly different at the 0.01 significance level (the p-value is less than 0.01 and the 99% confidence interval doesn't cover 0). However, we wouldn't recommend marketing this new drug: the new treatment is significantly *worse* than the current treatment.

One proportion

We can find confidence intervals and p-values for testing proportions using the function `prop.test`. Let's work through a simple example.

(Source: Openintro) Do people ever regret getting a tattoo? In a 2012 poll by Harris Interactive, 59 out of 423 respondents said yes. Based on the data in this study, find a 99% confidence interval for p, the proportion of people with tattoos who regret getting one.

As we mentioned, the command we need is `prop.test`. Here, the code we need is

```
prop.test(59, 423, conf.level = 0.99)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  59 out of 423, null probability 0.5
## X-squared = 218.48, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is not equal to 0.5
## 99 percent confidence interval:
##  0.1006219 0.1897715
## sample estimates:
##         p
## 0.1394799
```

First, we specify the number of events or "successes". Here, it's 59 because that's the number of people who regret getting a tattoo in our sample. Then, we specify the sample size, which in this case is 423. Finally, we specify our confidence level. If we didn't add `conf.level = 0.99`, `R` would assume we want 95% confidence. We can also see that the output provides a p-value for testing the hypothesis $H_0 : p = 0.5$ against $H_1 : p \neq 0.5$.

If we want to change the null value from 0.5 to, say, 0.2 (as an example) we would add in `p = 0.2` to our `prop.test` command. Also, if we wanted to change the alternative hypothesis from "not equal to" to "less" or "greater", we add that in using `alternative` in our `prop.test` statement. For the sake of giving a concrete example, if we wanted to test $H_0 : p = 0.2$ against $H_1 : p < 0.2$, we'd run:

```
prop.test(59, 423, p = 0.2, alternative = "less")
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  59 out of 423, null probability 0.2
## X-squared = 9.3087, df = 1, p-value = 0.00114
## alternative hypothesis: true p is less than 0.2
## 95 percent confidence interval:
##  0.0000000 0.1707564
## sample estimates:
##         p
## 0.1394799
```

The p-value is 0.00114, which is significant at the usual 0.05 threshold.

Two proportions

(Source: Openintro) In a study about online dating, 9 out of 40 males lied about their age and 5 out of 40 females lied about their age. Find a 95% confidence interval for the difference (% of men who lie about their age) - (% of women who lie about their age).

The function `prop.test` allows us to find confidence intervals and *p*-values for comparing two proportions. The code we need to solve this problem is:

```r
prop.test(c(9, 5), c(40, 40), conf.level = 0.95)
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(9, 5) out of c(40, 40)
## X-squared = 0.77922, df = 1, p-value = 0.3774
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.0900768  0.2900768
## sample estimates:
## prop 1 prop 2
##  0.225  0.125
```

First, we specify the number of events or "successes" in the two groups: here, we have 9 men and 5 women lying. Then, we specify the sample sizes of the groups, which is 40 for both men and women. Just as we had in the case where we only had one group, we can change the confidence level to, say, 99% confidence, by adding in `conf.level = 0.99` to the statement. We can also change the alternative hypothesis using `alternative` in the same way we saw for `prop.test`. For example, if we wanted our alternative hypothesis to be that men lie more than women, we'd write:

```r
prop.test(c(9, 5), c(40, 40), alternative = "greater")
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(9, 5) out of c(40, 40)
## X-squared = 0.77922, df = 1, p-value = 0.1887
## alternative hypothesis: greater
## 95 percent confidence interval:
##  -0.06353681  1.00000000
## sample estimates:
## prop 1 prop 2
##  0.225  0.125
```

The *p*-value is 0.1887, which is not significant at the 0.05 significance level.

χ^2-tests of independence for categorical variables

The `hsb2` dataset in `library(openintro)` has standardized test scores and background information for a sample of 200 high-schoolers. Let's read in the data:

```r
library(openintro)
data(hsb2)
```

If you want more information about the dataset, you can find it by typing `?hsb2`.

If we want to get a quick look at the variables in the dataset, we can use `str`:

```r
str(hsb2)
```

```
## tibble [200 x 11] (S3: tbl_df/tbl/data.frame)
##  $ id     : int [1:200] 70 121 86 141 172 113 50 11 84 48 ...
##  $ gender : chr [1:200] "male" "female" "male" "male" ...
##  $ race   : chr [1:200] "white" "white" "white" "white" ...
##  $ ses    : Factor w/ 3 levels "low","middle",..: 1 2 3 3 2 2 2 2 2 2 ...
##  $ schtyp : Factor w/ 2 levels "public","private": 1 1 1 1 1 1 1 1 1 1 ...
##  $ prog   : Factor w/ 3 levels "general","academic",..: 1 3 1 3 2 2 1 2 1 2 ...
##  $ read   : int [1:200] 57 68 44 63 47 44 50 34 63 57 ...
##  $ write  : int [1:200] 52 59 33 44 52 52 59 46 57 55 ...
##  $ math   : int [1:200] 41 53 54 47 57 51 42 45 54 52 ...
##  $ science: int [1:200] 47 63 58 53 53 63 53 39 58 50 ...
##  $ socst  : int [1:200] 57 61 31 56 61 61 61 36 51 51 ...
```

Suppose that a team of social scientists want to test whether the distribution of `ses` depends on `race`. First, we can create a table:

```r
table(hsb2$ses, hsb2$race)
```

```
##
##          african american asian hispanic white
##   low                  11     3        9    24
##   middle                6     5       11    73
##   high                  3     3        4    48
```

It's hard to see whether `race` depends on `ses` by looking at the raw counts. We can create columns with row and column percentages with `prop.table`. First, we save the original table with raw counts in a variable.

```r
tab = table(hsb2$ses, hsb2$race)
```

Then, we can find proportion tables with row and column percentages:

```r
round(prop.table(tab, 1), 3)
```

```
##
##          african american asian hispanic white
##   low               0.234 0.064    0.191 0.511
##   middle            0.063 0.053    0.116 0.768
##   high              0.052 0.052    0.069 0.828
```

```r
round(prop.table(tab, 2), 3)
```

```
##
##          african american asian hispanic white
##   low               0.550 0.273    0.375 0.166
##   middle            0.300 0.455    0.458 0.503
##   high              0.150 0.273    0.167 0.331
```

We can test whether `ses` and `race` are independent using a χ^2-test. The null hypothesis is that the variables are independent and the alternative hypothesis is that the variables are dependent. The code below runs a χ^2-test:

```r
chisq.test(tab)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tab
## X-squared = 18.516, df = 6, p-value = 0.005064
```

Note that the input is the table `tab`. The p-value is significant at the 0.05 significance level, so we reject the hypothesis that `ses` and `race` are independent at that significance level.

Pairwise comparisons of multiple normal means: Tukey HSD

Let's, once again, work with `hsb2` in `library(openintro)`.

Suppose that we want to compare the average scores in `math` among the levels of `race` doing pairwise tests, using Tukey HSD. The code is simple:

```r
mod = aov(math~race, data=hsb2)
TukeyHSD(mod)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = math ~ race, data = hsb2)
##
## $race
##                                  diff        lwr       upr     p adj
## asian-african american    10.5227273   1.838424 19.207030 0.0104479
## hispanic-african american  0.6666667  -6.337737  7.671071 0.9947149
## white-african american     7.2224138   1.704074 12.740754 0.0046342
## hispanic-asian            -9.8560606 -18.279657 -1.432465 0.0145446
## white-asian               -3.3003135 -10.535462  3.934835 0.6389480
## white-hispanic             6.5557471   1.457520 11.653975 0.0056430
```

We can check whether the pairwise differences are significant by checking the p-values. We can change the confidence level of the corrected intervals with the argument `conf.level`. For example, if we want 99% confidence intervals instead:

```r
TukeyHSD(mod, conf.level = 0.99)
```

```
##   Tukey multiple comparisons of means
##     99% family-wise confidence level
##
## Fit: aov(formula = math ~ race, data = hsb2)
##
## $race
##                                 diff          lwr        upr     p adj
## asian-african american    10.5227273  -0.04702112 21.0924757 0.0104479
## hispanic-african american  0.6666667  -7.85846043  9.1917938 0.9947149
## white-african american     7.2224138   0.50598962 13.9388380 0.0046342
## hispanic-asian            -9.8560606 -20.10849978  0.3963786 0.0145446
## white-asian               -3.3003135 -12.10628138  5.5056544 0.6389480
## white-hispanic             6.5557471   0.35064587 12.7608484 0.0056430
```