Confidence intervals in R. ANOVA

Alla Tambovtseva, Olga Lyashevskaya, Ilya Schurov, gRik Moroz

09-02-2019

Confidence intervals in R

Install library `DescTools` and load it:

```
# install.packages("DescTools")
```

```
library(DescTools)
```

Functions that we will use today:

- BinomCI -- binomial CI
- MeanCI -- CI for mean values
- MeanDiffCI -- CI for the difference in means

Confidence intervals for proportions

First, let us consider an abstract example so as to look at different effects connected with confidence intervals (sample size effect and confidence level effect). Suppose we tossed a coin 20 times and got 4 heads.

```
nheads <- 4 # number of heads
n1 <- 20  # total number of tosses
```

Confidence intervals for proportions

First, let us consider an abstract example so as to look at different effects connected with confidence intervals (sample size effect and confidence level effect). Suppose we tossed a coin 20 times and got 4 heads.

```
nheads <- 4 # number of heads
n1 <- 20  # total number of tosses
```

Now let's calculate a 95% confidence interval for the proportion of heads in such an experiment.

```
BinomCI(nheads, n1) # 95% by default
```

Calculate the length of a confidence interval:

```
ci.95 <- BinomCI(nheads, n1)
ci.95[3] - ci.95[2]
```

Calculate the length of a confidence interval:

```
ci.95 <- BinomCI(nheads, n1)
ci.95[3] - ci.95[2]
```

Increase the number of tosses (number of heads remains the same):

```
n2 <- 40 # now 40 tosses
ci.95.2 <- BinomCI(nheads, n2)
ci.95.2[3] - ci.95.2[2]
```

Calculate the length of a confidence interval:

```
ci.95 <- BinomCI(nheads, n1)
ci.95[3] - ci.95[2]
```

Increase the number of tosses (number of heads remains the same):

```
n2 <- 40 # now 40 tosses
ci.95.2 <- BinomCI(nheads, n2)
ci.95.2[3] - ci.95.2[2]
```

It shrinked, right?

Calculate the length of a confidence interval:

```
ci.95 <- BinomCI(nheads, n1)
ci.95[3] - ci.95[2]
```

Increase the number of tosses (number of heads remains the same):

```
n2 <- 40 # now 40 tosses
ci.95.2 <- BinomCI(nheads, n2)
ci.95.2[3] - ci.95.2[2]
```

It shrinked, right?

Keep the number of tosses equal to 20, but increase the confidence level:

```
ci.99 <- BinomCI(nheads, n1, conf.level = 0.99)
ci.99[3] - ci.99[2]
```

Calculate the length of a confidence interval:

```
ci.95 <- BinomCI(nheads, n1)
ci.95[3] - ci.95[2]
```

Increase the number of tosses (number of heads remains the same):

```
n2 <- 40 # now 40 tosses
ci.95.2 <- BinomCI(nheads, n2)
ci.95.2[3] - ci.95.2[2]
```

It shrinked, right?

Keep the number of tosses equal to 20, but increase the confidence level:

```
ci.99 <- BinomCI(nheads, n1, conf.level = 0.99)
ci.99[3] - ci.99[2]
```

It extended.

Let's set a true probability of getting a head in one toss of a coin.

```
p0 <- 0.5 # true probability of getting a head in one tossing
```

Then take 1000 samples of size 100, calculate confidence intervals for proportion of ones in each sample and count how many intervals contain a population proportion (the true probability of getting a head in one toss of a coin).
Recall the code from our previous seminars and suppose we asked 1000 people to toss a coin 100 times and report the proportion of heads they obtained.

```
n <- 100
samples <- 1000
dat <- matrix(sample(c(0, 1),
                     n * samples,
                     replace=TRUE), ncol=n, byrow=TRUE)
```

Let's set a true probability of getting a head in one toss of a coin.

```
p0 <- 0.5 # true probability of getting a head in one tossing
```

Then take 1000 samples of size 100, calculate confidence intervals for proportion of ones in each sample and count how many intervals contain a population proportion (the true probability of getting a head in one toss of a coin).
Recall the code from our previous seminars and suppose we asked 1000 people to toss a coin 100 times and report the proportion of heads they obtained.

```
n <- 100
samples <- 1000
dat <- matrix(sample(c(0, 1),
                     n * samples,
                     replace=TRUE), ncol=n, byrow=TRUE)
```

```
cis <- BinomCI(rowSums(dat), n)
colnames(cis) <- c("est", "lower", "upper")

head(cis[,"lower"])
head(cis[,"upper"])

sum(cis[,2] <= p0 & cis[,3] >= p0)
```

5 / 20

Confidence intervals: real data

Now let's proceed to real data and work with *Verses* data set.

```
verses <- read.csv("https://raw.githubusercontent.com/LingData2019/LingData/ma
str(verses) # recall which variables are there
```

Confidence intervals: real data

Now let's proceed to real data and work with *Verses* data set.

```
verses <- read.csv("https://raw.githubusercontent.com/LingData2019/LingData/ma
str(verses) # recall which variables are there
```

Calculate a confidence interval for the proportion of nouns at the end of lines:

```
nnouns <- nrow(verses[verses$UPoS == "NOUN", ])
total <- nrow(verses)

BinomCI(nnouns, total)
```

Confidence intervals for means

Now let's turn to the data set on Icelandic language from our previuos class.

```
phono <- read.csv("http://math-info.hse.ru/f/2018-19/ling-data/icelandic.csv")
```

Confidence intervals for means

Now let's turn to the data set on Icelandic language from our previuos class.

```
phono <- read.csv("http://math-info.hse.ru/f/2018-19/ling-data/icelandic.csv")
```

Choose aspirated and non-aspirated cases again:

```
asp <- phono[phono$aspiration == "yes", ]
nasp <- phono[phono$aspiration == "no", ]
```

Confidence intervals for means

Now let's turn to the data set on Icelandic language from our previuos class.

```
phono <- read.csv("http://math-info.hse.ru/f/2018-19/ling-data/icelandic.csv")
```

Choose aspirated and non-aspirated cases again:

```
asp <- phono[phono$aspiration == "yes", ]
nasp <- phono[phono$aspiration == "no", ]
```

Calculate confidence intervals for mean values of vowel duration in each group:

```
MeanCI(asp$vowel.dur)
MeanCI(nasp$vowel.dur)
```

Plot them using `sciplot`:

```
# install.packages("sciplot")
library(sciplot)
lineplot.CI(data = phono,
            response = vowel.dur,
            x.factor = aspiration)
```

Confidence intervals and significance of differences

- If two CI's for a population parameter (proportion, mean, median, etc) do not overlap, it means that true values of population parameters are significantly different.
- If two CI's for a population parameter overlap, true values of population parameters can (to be equal to each other), but **not** necessarily do so. For example, if two confidence intervals for means overlap, we cannot make a definite conclusion, more accurate testing is required (t-test). So, in general, comparison of confidence intervals (with the same confidence level, of course) is **not** equivalent to hypotheses testing.

Confidence intervals and significance of differences

- If two CI's for a population parameter (proportion, mean, median, etc) do not overlap, it means that true values of population parameters are significantly different.
- If two CI's for a population parameter overlap, true values of population parameters can (to be equal to each other), but **not** necessarily do so. For example, if two confidence intervals for means overlap, we cannot make a definite conclusion, more accurate testing is required (t-test). So, in general, comparison of confidence intervals (with the same confidence level, of course) is **not** equivalent to hypotheses testing.

Consider a case when two CI's for means overlap, but population means are significantly different. Let's select only cases with aspirated consonants and compare the average vowel duration for round and unrounded vowels.

```
w1 <- phono[phono$aspiration == 'yes' & phono$roundness == "round", ]
w2 <- phono[phono$aspiration == 'yes' & phono$roundness == "unrounded", ]
```

Do CI's overlap?

```
MeanCI(w1$vowel.dur)
MeanCI(w2$vowel.dur)
```

Can we conclude that mean vowel duration is different for round and unrounded vowels?

T-test

Perform a statistical test, a two sample Student's t-test.

```
# reject or not reject H0
t.test(w1$vowel.dur, w2$vowel.dur)
```

T-test

Perform a statistical test, a two sample Student's t-test.

```
# reject or not reject H0
t.test(w1$vowel.dur, w2$vowel.dur)
```

Null hypotheses should be rejected, so population means are different.

T-test

Perform a statistical test, a two sample Student's t-test.

```
# reject or not reject H0
t.test(w1$vowel.dur, w2$vowel.dur)
```

Null hypotheses should be rejected, so population means are different.

Actually, testing hypothesis about the equality of population means is equivalent to finding whether *a CI for the difference of means* includes zero.

```
# CI for difference between means
MeanDiffCI(w1$vowel.dur, w2$vowel.dur)
```

T-test

Perform a statistical test, a two sample Student's t-test.

```
# reject or not reject H0
t.test(w1$vowel.dur, w2$vowel.dur)
```

Null hypotheses should be rejected, so population means are different.

Actually, testing hypothesis about the equality of population means is equivalent to finding whether *a CI for the difference of means* includes zero.

```
# CI for difference between means
MeanDiffCI(w1$vowel.dur, w2$vowel.dur)
```

Thus, intersection of CI's for means (or for any population parameters) $\neq$ CI for the difference includes zero $\neq$ H_0 about equality should not be rejected.

ANOVA

Load data on Icelandic:

```
phono <- read.csv("http://math-info.hse.ru/f/2018-19/ling-data/icelandic.csv")
```

```
str(phono)
```

```
## 'data.frame':    806 obs. of  31 variables:
##  $ speaker        : Factor w/ 5 levels "brs02","bte03",..: 1 1 1 2 2 2 3 3 3 4 ...
##  $ index          : int  137 138 139 15 16 17 4 5 6 113 ...
##  $ word           : Factor w/ 58 levels "detta","dögg",..: 29 29 29 29 29 29 29 29
##  $ time           : num  438 441 443 133 138 ...
##  $ word.dur       : num  489 444 450 530 515 ...
##  $ voicing.dur    : num  153.2 138.7 169.8 93.6 159.3 ...
##  $ vowel.dur      : num  100.9 72.6 107.5 93 107.4 ...
##  $ cluster.dur    : num  231 217 195 245 230 ...
##  $ spreading.dur  : num  114.3 98.2 50.1 119.2 45.1 ...
##  $ sonorant.dur   : num  167 164 112 120 97 ...
##  $ closure.dur    : num  64.7 52.7 82.3 125.3 133.2 ...
##  $ vor            : num  332 290 302 338 338 ...
```

Create a boxplot for vowel duration for each group of consonants:

```
boxplot(phono$vowel.dur ~ phono$cons1)
```

Perform ANOVA to find out if the variation between groups is significantly larger than the variation within groups:

```
res <- aov(phono$vowel.dur ~ phono$cons1)
res
```

This type of ANOVA is called a one-way ANOVA for independent groups.

Perform ANOVA to find out if the variation between groups is significantly larger than the variation within groups:

```
res <- aov(phono$vowel.dur ~ phono$cons1)
res
```

This type of ANOVA is called a one-way ANOVA for independent groups. More informative summary:

```
# H0: there are no difference in population means by groups
summary(res)
```

ANOVA with multiple groups:

```
res <- aov(phono$vowel.dur ~ phono$cons1 + phono$roundness)
res
```

ANOVA with multiple groups:

```
res <- aov(phono$vowel.dur ~ phono$cons1 + phono$roundness)
res
```

More informative summary:

```
# H0: there are no difference in population means by groups
summary(res)
```

Create a boxplot for vowel duration for each group of consonants and for both groups of roundness:

```
boxplot(phono[phono$roundness == "round",]$vowel.dur ~ phono[phono$roundness =
```

Create a boxplot for vowel duration for each group of consonants and for both groups of roundness:

```
boxplot(phono[phono$roundness == "unrounded",]$vowel.dur ~ phono[phono$roundne
```

Plot all groups together:

```
boxplot(phono$vowel.dur ~ phono$cons1 + phono$roundness)
```

Create a boxplot for vowel duration for each group of consonants and for both groups of roundness:

```
boxplot(phono[phono$roundness == "unrounded",]$vowel.dur ~ phono[phono$roundne
```

Create the same boxplot with ggplot2

```
ggplot(data = phono, # add the data
       aes(x = cons1, y = vowel.dur, # set x, y coordinates
           color = cons1)) +    # color by cons1
  geom_boxplot() +
  facet_grid(~roundness) # create panes base on health status
```