Linguistic Data: Quantitative Analysis and Visualisation

Confidence intervals in R

Ilya Schurov, Olga Lyashevskaya, George Moroz, Alla Tambovtseva

09 February 2018

Install library `DescTools` and load it:

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

```
library(DescTools)
```

Confidence intervals for proportions

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

```
nheads1 <- 4 # number of heads
n1 <- 20  # total number of tosses
```

What is the probability of getting a head in one tossing? We do not know it exactly since we know nothing about the features of our coin (at least, whether it is fair or not). However, we can calculate a confidence interval for it.

Now let's calculate a 95% confidence interval for the probability of obtaining a head in one toss of a coin (proportion of heads in such an experiment).

```
BinomCI(nheads1, n1) # 95% CI by default
```

```
##      est     lwr.ci    upr.ci
## [1,] 0.2 0.08065766 0.4160174
```

Calculate the length of a confidence interval:

```
ci.95 <- BinomCI(nheads1, n1)
ci.95[3] - ci.95[2]
```

```
## [1] 0.3353598
```

Let's increase the number of heads and the number of tosses (the proportion of heads remains the same):

```
nheads2 <- 40
n2 <- 200 # now 200 tosses
ci.95.2 <- BinomCI(nheads2, n2)
ci.95.2[3] - ci.95.2[2]  # it shrinked
```

```
## [1] 0.1104032
```

The confidence interval has become narrower. And the ratio of the lengths of two confidence intervals should be $\sqrt{N}$ approximately, where N is a number of times we increase the sample size. In our case it is 10 (from 20 to 200).

```
sqrt(10) # square root of N
```

```
## [1] 3.162278
```

```
0.3353598/0.1104032 # ratio of lengths
```

```
## [1] 3.037591
```

Now let's keep the number of tosses equal to 200, but increase the confidence level:

```
ci.99 <- BinomCI(nheads2, n2, conf.level = 0.99)
ci.99[3] - ci.99[2] # it extended
```

```
## [1] 0.1446413
```

Now let's try to 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).

Now 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.

```
tosses <- 100 # series of 100 tosses (for one person)
samples <- 1000  # 1000 people tossed a coin
dat <- matrix(sample(c(0, 1),
                     tosses * samples,
                     replace=TRUE), ncol=tosses, byrow=TRUE)
# recall how dat looks like
head(dat, 2)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]    0    1    0    1    1    0    1    0    0     1     0     0     1
## [2,]    0    1    1    0    0    0    1    0    0     1     1     0     0
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## [1,]     1     0     0     0     0     1     1     1     1     0     0
## [2,]     1     1     0     0     1     0     1     1     0     1     1
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35]
## [1,]     0     1     0     0     1     1     0     1     0     1     1
## [2,]     1     1     0     0     1     0     1     0     1     1     0
##      [,36] [,37] [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46]
## [1,]     0     0     1     1     1     1     1     1     0     1     1
## [2,]     1     1     1     0     0     1     1     1     0     0     0
##      [,47] [,48] [,49] [,50] [,51] [,52] [,53] [,54] [,55] [,56] [,57]
## [1,]     1     0     1     1     1     0     1     0     0     0     0
## [2,]     1     0     0     0     1     1     1     1     0     1     0
##      [,58] [,59] [,60] [,61] [,62] [,63] [,64] [,65] [,66] [,67] [,68]
## [1,]     1     0     1     1     0     0     0     0     1     0     0
## [2,]     0     0     1     0     0     0     0     0     0     0     0
##      [,69] [,70] [,71] [,72] [,73] [,74] [,75] [,76] [,77] [,78] [,79]
## [1,]     1     0     0     0     0     1     1     0     0     1     0
## [2,]     1     1     1     0     0     1     1     0     0     0     1
##      [,80] [,81] [,82] [,83] [,84] [,85] [,86] [,87] [,88] [,89] [,90]
## [1,]     0     1     0     1     1     1     1     0     1     0     1
## [2,]     0     0     1     0     1     1     1     0     1     0     1
##      [,91] [,92] [,93] [,94] [,95] [,96] [,97] [,98] [,99] [,100]
## [1,]     1     1     0     0     0     0     0     1     1      1
## [2,]     1     0     0     0     0     0     1     0     1      0
```

Now calculate confidence intervals for the probability of getting a head in one toss based on proportions on heads in each series of tosses (based on each row in `dat`):

```
cis <- BinomCI(rowSums(dat), tosses)
head(cis)
```

```
##      est    lwr.ci    upr.ci
## x.1 0.50 0.4038315 0.5961685
## x.2 0.46 0.3656081 0.5573514
## x.3 0.52 0.4231658 0.6153545
## x.4 0.46 0.3656081 0.5573514
## x.5 0.47 0.3751082 0.5671114
## x.6 0.49 0.3942200 0.5865199
```

So as to decide how many confidence intervals include true population proportion `p0` (probability of getting a head in one toss). To do so we need the second and the third column of `cis`:

```
head(cis[,"lwr.ci"])
```

```
##       x.1       x.2       x.3       x.4       x.5       x.6
## 0.4038315 0.3656081 0.4231658 0.3656081 0.3751082 0.3942200
```

```
head(cis[,"upr.ci"])
```

```
##       x.1       x.2       x.3       x.4       x.5       x.6
## 0.5961685 0.5573514 0.6153545 0.5573514 0.5671114 0.5865199
```

Now we can check whether each confidence interval includes `p0`. If it is true, `p0` should be greater or equal to the lower bound of an interval and less or equal to the upper bound.

```
head(cis[,"lwr.ci"] <= p0 & cis[,"upr.ci"] >= p0)
```

```
##  x.1  x.2  x.3  x.4  x.5  x.6
## TRUE TRUE TRUE TRUE TRUE TRUE
```

```
# count the proportion of CIs that include p0
mean(cis[,"lwr.ci"] <= p0 & cis[,"upr.ci"] >= p0)
```

```
## [1] 0.945
```

It is approximately 0.95 as expected.

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/master/data/poetry_last_in_
#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)
```

```
##            est    lwr.ci    upr.ci
## [1,] 0.6098901 0.5588825 0.6586025
```

Confidence intervals for means

Now let's work with 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)
```

```
##     mean   lwr.ci   upr.ci
## 78.75772 76.68274 80.83270
```

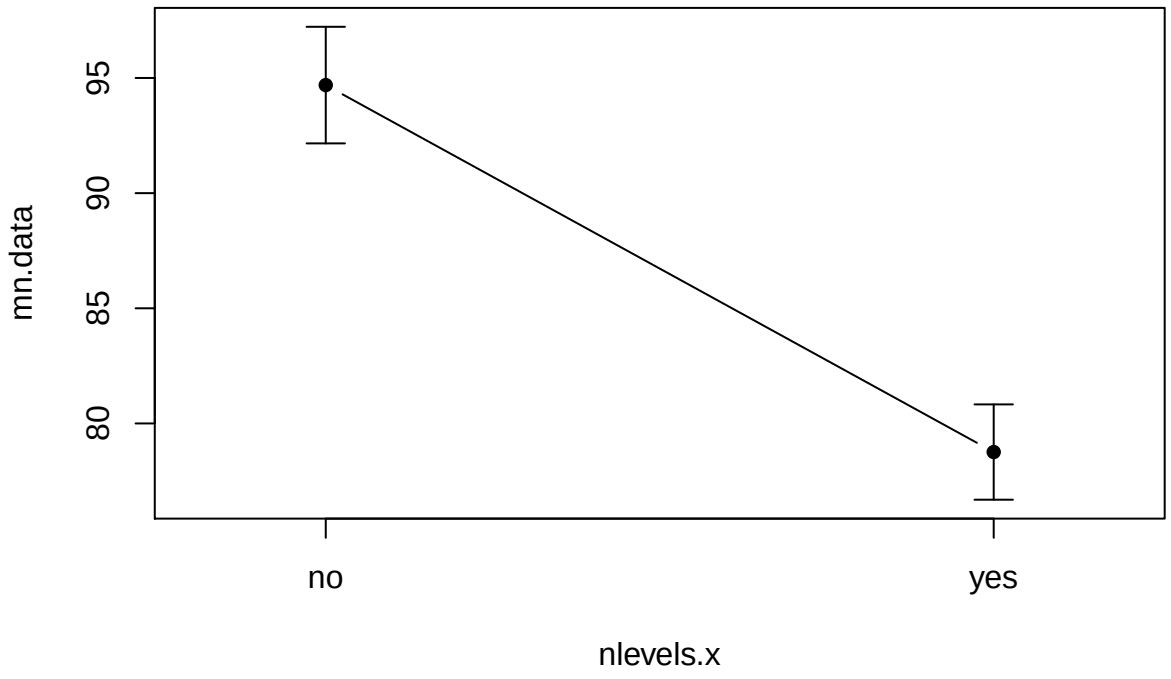
```
MeanCI(nasp$vowel.dur)
```

```
##     mean   lwr.ci   upr.ci
## 94.69124 92.15292 97.22957
```

Plot them using `sciplot`:

```
install.packages("sciplot")
```

```
library(sciplot)
# specify data
# response is a variable for which mean we plot a CI
# x.factor is a grouping variable (as we create plots by groups)
# ci.fun - function that calculates CI (1.96 multipled by standard error)
lineplot.CI(data = phono,
            response = vowel.dur,
            x.factor = aspiration,
            ci.fun = function(x) c(mean(x)-1.96*se(x), mean(x)+1.96*se(x)))
```



Bold dots here correspond to sample means and whiskers (called error bars) correspond to the bounds of confidence intervals for means. From this graphs we can see, for example, whether confidence intervals overlap. Why it can be helpful, we will discuss right now.

Confidence intervals and statistical 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 coincide (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)
```

```
##     mean   lwr.ci   upr.ci
## 81.74052 77.89567 85.58537
```

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

```
##     mean   lwr.ci   upr.ci
## 76.90839 74.54499 79.27179
```

They overlap! Can we conclude that mean vowel duration is different for round and unrounded vowels? In fact, no. Let us see.

Now perform an accurate test, a two sample Student's t-test.

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

```
##
##  Welch Two Sample t-test
##
## data:  w1$vowel.dur and w2$vowel.dur
## t = 2.1134, df = 269.27, p-value = 0.03549
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.3304964 9.3337590
## sample estimates:
## mean of x mean of y
##  81.74052  76.90839
```

Null hypotheses should be rejected, so population means are different. So, this is an illustration of the fact described above: two confidence intervals overlap, but population means are statistically different.

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

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

```
##  meandiff    lwr.ci    upr.ci
## 4.8321277 0.3304964 9.3337590
```

So, 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.