Introduction to Statistical Tests

Marcin Kierczak

2017-10-24

Introduction

Statistics is a branch of science, more precisely a branch of mathematics that is concerned with collection, analysis, interpretation, and presentation of data. A very similar term, *statistic* denotes a single measure, a number that is describing in a concise way some feature of a data sample. Statistic is usually a number resulting from application of a statistical procedure. Examples of a statistic include, sample mean and standard deviation of a sample.

Basic statistics and statistical terms

Here, we introduce some fundamental concepts used in statistics. We begin by describing distributions and samples. Distributions are described by their *parameters* while samples are described by *statistics*. Every distribution can be described by parameters belonging to two classes: *location parameters* and *dispersion parameters*. Location parameters include *mean* and *median* while dispersion parameters include *variance* and *standard deviation*.

Mean

Mean is a location parameter measuring the central tendency of the data. It is defined as a sum of all the values divided by the number of observations:

$$\mu = \frac{\sum x}{N}$$

Population mean is denoted as μ while sample mean as $\bar{X}$. Here, N is the number of observations and $\sum x$ is the sum of all observed values.

MEAN CAN BE MISLEADING when a distribution is skewed (non-symmetrical) and can be greatly influenced by outliers. To remedy the latter problem, other types of mean such as *weighted mean* or a mean that does not take into account the extreme observations called the *Winsor mean* are used. We, however, will not discuss these here.

Median

Median is another measure of the central tendency or a location parameter of a distribution. Unlike the mean, it is not taking into account all observations. Median is simply the middle value in the or-

dered data series. If there is an odd number of data points, median is easy to find. If there is an even number of data points, median is defined as the mean of the two data points in the middle.

BY COMPARING MEAN WITH MEDIAN, we can tell something about how skewed the distribution is. For a perfectly symmetrical distribution median and mean are equal. For *left-skewed* distribution, median is less than the mean. For *right-skewed* distributions, median is greater than the mean.

Variance

Population variance is denoted as σ^2 and sample variation as s. Variance measures dispersion of observations around the mean. The more spread they are, the higher value of the variance. Population variance is defined as:

$$\sigma^2 = \frac{\sum (x - \mu)^2}{n}$$

while sample variation is:

$$s^2 = \frac{\sum (x - \bar{x})^2}{n - 1}$$

WHY ARE WE DIVIDING BY $(n-1)$ instead of dividing by n? We want our sample variance s^2 to be as accurate estimate of population variance σ^2 as possible. Imagine that we are estimating variance in weight of sheep in a herd (population) of 2000 individuals using a sample of 200 animals. It is very likely that we will miss in our sample those few light or heavy individuals which would have, otherwise, influenced our estimation of variance quite a lot (since they give very large $(x-\bar{x})^2$ term). Therefore we correct for this byusing $n-1$ instead of n. Observe that for very small sample sizes subtraction of 1 has a large effect on variance while for large sample sizes it has very minute effect:

```
# A function to compute population variance.
pop.var <- function(x) {
  sq.dev <- (x - mean(x))^2
  pop.var <- sum(sq.dev) / length(x)
  return(pop.var)
}
# We generate a population of sheep
# True population variance is 16
# so sd=4
pop <- rnorm(n=2000, mean=55, sd=4)
# Number of sampling events per sample size
```

```
N <- 1000

compare.var <- function(pop, N, sample.size) {
  s <- array(dim=N) # Sample variance (with correction)
  sigma <- array(dim=N) # Population variance
  for (i in 1:N) {
    my.sample <- sample(pop, size=sample.size, replace=F)
    s[i] <- var(my.sample)
    sigma[i] <- pop.var(my.sample)
  }
  result <- c(median(s), median(sigma))
  return(result)
}

# Actual simulation
sample.sizes <- c(5:100)
result <- c()
for (sample.size in sample.sizes) {
  tmp <- compare.var(pop, N, sample.size)
  new <- cbind(sample.size, s=tmp[1], sigma=tmp[2])
  result <- rbind(result, new)
}
result <- as.data.frame(result)
tmp <- c(result$s, result$sigma)
```

You can perhaps see that for small sample sizes, the effect of using $n-1$ instead of 1 is much larger!

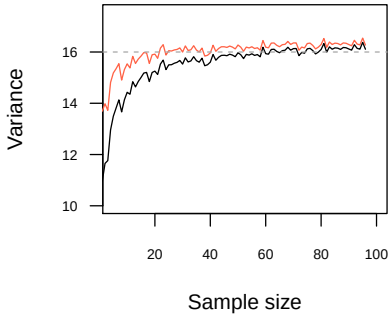


Figure 1: The effect of using $n-1$ in denominator of variance estimation. Black dots – no correction, red dots – corrected. True population variance – darkgrey dotted line.

Note that variance cannot take negative values, simply because we are squaring the numerator. Often, a square root of variance called *standard deviation* is used instead of variance itself. Standard deviation (sd, SD, std.dev.) is denoted by σ for population and by s for sample.

If two or more traits are measured using different scales, variances cannot be directly compared due to scale effect. Therefore, often variables to be compared are *standardized* so that they follow the *standard* normal distribution $\mathcal{N}(0,1)$ with $\bar{X}=0$ and $s=1$. Standardization maps the actual values into the corresponding *z-scores* which tell how many standard deviations away from the mean a given observation is: $z_{score}=(x-\bar{x})/s$.

Covariance

When looking at two variables, say x and y, it is often interesting to know how similar the deviations from the mean of one variable are to the deviations of the other variable. This is measured by *covariance*:

$$Cov_{x,y} = \frac{\sum (x - \bar{x}) \cdot (y - \bar{y})}{n - 1}$$

OBSERVE THAT VARIANCE is a covariance of a variable with itself – substitute $(y - \bar{y})$ with $(x - \bar{x})$ to get variance. Similarly to variance, covariance is scale dependent! Positive covariance means that y increases with x, for the opposite situation, we have negative covariance. When the two variables are *orthogonal* to each other (independent), covariance equals zero.

Correlation

As mentioned above, covariance is scale dependent. It can, however, easily be re-sacled to be bound between -1 and 1. This operation is analoguous to standardization. Such re-scaling yields *correlation*:

$$Cor_{x,y} = \frac{Cov_{x,y}}{\sqrt{s_x^2 \cdot s_y^2}}$$

To see what is the scale effect on covariance, look at the following example:

```
w <- rnorm(100, mean=0, sd=1)
x <- 4 * jitter(w)
cov(w, x)
```

```
## [1] 3.658909
```

```
cor(w, x)
```

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

```
y <- rnorm(100, mean=0, sd=10)
z <- 4 * jitter(y)
cov(y, z)
```

```
## [1] 458.0993
```

```
cor(y, z)
```

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

Statistical tests

Here, we will discuss some most common and useful statistical tests. First, there is a very important distinction between two types of statistical tests:

- **parametric tests** – they assume certain parameters of the population and sample distribution. Usually these have to be normal and μ and σ have to be known.
- **non-parametric tests** – they do not make assumptions about the distributions.

Then – one may ask – why not to use non-parametric tests only? The answer is that typically the parametric tests have higher power than the non-parametric ones.

One-sample tests

First, let us sonsider tests where we are considering one sample. The very first step is to check normality of the sample distribution using, e.g. Shapiro-Wilk test:

```
data <- rnorm(n=100, mean=0, sd=1)
shapiro.test(data)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data
## W = 0.98046, p-value = 0.1443
```

First, we generated a sample of 100 random numbers coming from the normal distribution $\mathcal{N}(0,1)$ with mean equal to 0 and standard deviation equal to 1. Next, we performed the Shapiro-Wilk normality test. Our null hypothesis is that H_0 : *distribution is normal*. P-value obtained from the test is much greater than 0.05 which[1] does not give us a reason to reject the null hypothesis. Thus, we can say that our sample is not significantly departing from normality. Now, let us see what happens if we try to test a uniform distribution using Shapiro-Wilk test:

[1] using significance level $\alpha = 5\%$

```
data2 <- runif(100, min = 2, max = 4)
shapiro.test(data2)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  data2
## W = 0.96413, p-value = 0.008011
```

NOW, P-VALUE IS LESS THAN 0.05 and gives us a reason to reject the null. We can suspect the distribution is departing from normality. Finally, let us compare the two samples using a **quantile-quantile** (Q-Q) plot:

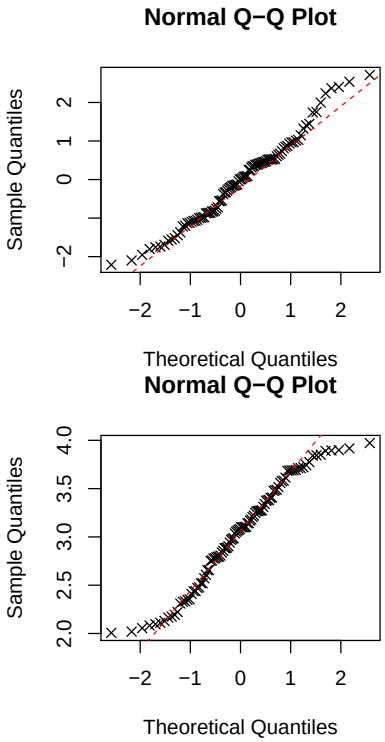


Figure 2: Q-Q plots for `data` (top) and `data2` (bottom). Data points on the left panel clearly follow the straight line which is a sign of normality. This is not the case for datapoints on the right panel. Shapiro-Wilk tests confirmed this observation.

AT THE NEXT STEP, one may wish to check whether the sample comes from a population characterized by a given mean $\bar{X}$. This can be accomplished with a simple Student's t-test.[2]

```
t.test(data, mu=1.1)
```

```
##
##  One Sample t-test
##
## data:  data
## t = -10.391, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 1.1
## 95 percent confidence interval:
##  -0.2515569  0.1818681
## sample estimates:
##  mean of x
## -0.0348444
```

AS WE CAN SEE, very small value of p-value lets us reject our nill hypothesis H_0 : *sample comes from population with mean* $\bar{X} = 1.1$. We can also see that the 95% confidence interval is between -0.294 and 0.144, i.e. the population mean from which the sample comes is somewhere in this interval.

WE CAN ALSO ASK ANOTHER TYPE OF QUESTION: does our sample come from a population with a given variance? To answer this question, we will use the Z test:

```
critical <- 0.05
sigma <- 3
conf <- qnorm(1 - 0.5 * critical)
std.err <- sigma/sqrt(length(data))
conf.interval = mean(data) + c(-std.err * conf, std.err * conf)
```

HERE, WE A PRIORI KNEW that variance in the population is $\sigma = 3$. Now, if the population mean μ is within the determined confidence interval, we can conclude that sample comes from this population $\mathcal{N}(\mu, 3)$.

[2] The t-statistics and t-test were invented by William Sealy Gosset. He was an employee of Guiness brewery and invented this test to monitor quality of their famous stout.

FINALLY, let us use the non-parametric Wilcoxon test to see whether our sample coming from the uniform distribution comes from a population with mean $\mu = 3$.

```
wilcox.test(data2, mu=3)
```

```
##
##  Wilcoxon signed rank test with
##  continuity correction
##
## data:  data2
## V = 2752, p-value = 0.4361
## alternative hypothesis: true location is not equal to 3
```

As we can see, there is no reason to reject our null hypothesis: H_0 : *sample comes from a population with* $\mu = 3$.

Two-sample tests

In this section we will consider two samples. We want to know whether they come from the same population. We begin by testing whether our two samples have homogenous variance. This can be done using Sendecor F-test:

```
data3 <- rnorm(100, 0.2, 3)
data4 <- rnorm(100, 0.001, 1)
var.test(data, data3)
```

```
##
##  F test to compare two variances
##
## data:  data and data3
## F = 0.094207, num df = 99, denom df =
## 99, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.0633866 0.1400141
## sample estimates:
## ratio of variances
##          0.0942073
```

```
var.test(data, data4)
```

```
##
##  F test to compare two variances
##
## data:  data and data4
```

```
## F = 1.4301, num df = 99, denom df = 99,
## p-value = 0.07665
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9622111 2.1254193
## sample estimates:
## ratio of variances
##           1.430071
```

As we could expect (we know parameters of distributions used to generate the data), *data* and *data3* do not have homogenous variances while *data* and data *data4* have. Now, we want to ask a question whether *data* and *data4* come from the same population:

```
t.test(data, data4)
```

```
##
##  Welch Two Sample t-test
##
## data:  data and data4
## t = -0.81119, df = 191.99, p-value =
## 0.4183
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.3963058  0.1653238
## sample estimates:
##  mean of x  mean of y
## -0.0348444  0.0806466
```

Apparently, there is no reason to think (at $\alpha = 0.95$) that the two samples come from different populations. For `data` and `data3`, we cannot apply t-test since the variance was not homogenous. We need its non-parametric counterpart, U-Mann-Whitney test (implemented in `wilcox.test`):

```
wilcox.test(data, data3)
```

```
##
##  Wilcoxon rank sum test with continuity
##  correction
##
## data:  data and data3
## W = 4325, p-value = 0.09934
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(data, data2)
```

```
##
##  Wilcoxon rank sum test with continuity
##  correction
##
## data:  data and data2
## W = 93, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Results are to be interpreted in the same way as in the previous tests.

Beyond two-sample tests

What if we have to compare more than two samples? Certinly, one can perform a number of pairwise two-sample tests, but there are other ways of doing it. First, homogeneity of variance for many samples can be tested using Bartlett's test or Levene's test:

```
data.new <- data.frame(data=c(data, data2, data3),
                       group=rep(1:3,each=length(data)))
bartlett.test(data~group, data.new)
```

```
##
##  Bartlett test of homogeneity of
##  variances
##
## data:  data by group
## Bartlett's K-squared = 311.38, df = 2,
## p-value < 2.2e-16
```

```
library(car)
with(data.new, leveneTest(data, as.factor(group)))
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   2  93.599 < 2.2e-16 ***
##       297
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
```

We can see that both tests give similar answer – there is no variance homogeneity between these three samples. This implies the use of a non-parametric test to check whether all three samples come from a population with a given mean. We will use Kruskal-Wallis test:

```
kruskal.test(data ~ group, data.new)
```

In this case, apparently not all the samples come from the same population with the same mean.

Usefulness of χ^2 tests

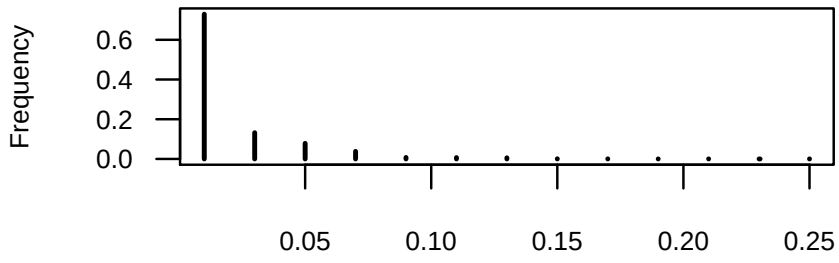
Below, we will hava a closer look at a class of very useful tests based on χ^2 statistic. In general, χ^2 tests are used when we consider counts or proportions.

$$\chi^2 = \frac{(N_{exp} - N_{obs})^2}{N_{exp}}$$

The χ^2 distribution is also relatively simple to derive by using simulations in R. Let us assume a population with known ratio of two classes, e.g. population of a poll respondents who answered "yes" or "no"" no to a particular question. We know the ratio of these categories in our population and start drawing samples of a given size from the population. We would like to know how many times the "yes" to "no" ratio in the sample matches the "yes" to "no" ratio in the population. In the simulation below, we encode "yes" and "no" as *TRUE* and *FALSE* respectively.

```
# A function to simulate chi square distribution
simulate.chi.sq <- function(pop, probs=c(0.5, 0.5),
                            sample.size=NULL) {
  if (is.null(sample.size)) {
    # By default sample size goes
    # to 10% the population size
    sample.sizes <- c(1:floor(.1 * pop.size))
  }
  sample.ind <- sample(1:1000, size=sample.size, rep=F)
  my.sample <- pop[sample.ind]
  exp <- sum(pop)/length(pop)
  obs <- sum(my.sample)/length(my.sample)
  chi.sq <- (exp - obs)^2/exp
  chi.sq
}

# Simulate a population of 1000 individuals with 50:50 ratio.
pop <- sample(c(T,F), size=1000, replace=T, prob=c(.5,.5))
# Run simulation N times
N <- 10000
result <- NULL
for (i in 1:N) {
  result <- rbind(result,
                  simulate.chi.sq(pop=pop,
                                  probs=c(0.5, 0.5),
                                  sample.size=30))
}
```



The above simulation works as follows: first, we create a population of "yes" and "no" values, with the probability of each value equal to 0.5. Next, we sample 30 individuals and compute χ^2 value for them. We repeat such sampling 10 000 times and plot the frequency of χ^2 values. As you can see, the vast majority of the values is close to zero which we expected: proportion of "yes" to "no" in the sample should be close to the true proportion in the population. Based on this distribution, we can set a confidence threshold so that only 5% of the χ^2 values are to the right of it. Thus, if we get our χ^2 value above the threshold, we can reject our H_0 : *proportion in the sample is the same as in the population* and be wrong only 5% of the times…

Tests for proportion

Say we have examined a Petri plate that was exposed for the environment of the main hall of our University for some time, than incubated for 40h at 37C. We have counted 100 colonies of *Escherichia coli* and 30 colonies of *Staphylococcus epidermidis*. Does it give us reason to say that *S. epidermidis* constitutes 25% of the population of bacteria in the main hall? Disregard all the factors like other bacteria that may grow optimally at a different temperature, lack of replicates etc. We will use a χ^2 test:

```
prop.test(30, 100, p=0.25)
```

```
##
##  1-sample proportions test with
##  continuity correction
##
## data:  30 out of 100, null probability 0.25
## X-squared = 1.08, df = 1, p-value =
## 0.2987
## alternative hypothesis: true p is not equal to 0.25
## 95 percent confidence interval:
##  0.2145426 0.4010604
## sample estimates:
```

```
##   p
## 0.3
```

Apparently, we can say so. However, given the observed count, at $\alpha = 95\%$ the real proportion may be somewhere between 21% and 40%.

Well, this is also a handy tool when newspapers start serving us exit-poll results. If they, say, write that they tested 100 people and 55 declared support for The Only Right candidate does it mean (s)he will win?

```
prop.test(55, n=100)
```

```
##
##  1-sample proportions test with
##  continuity correction
##
## data:  55 out of 100, null probability 0.5
## X-squared = 0.81, df = 1, p-value =
## 0.3681
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.4475426 0.6485719
## sample estimates:
##    p
## 0.55
```

Well, it gives them right to say that the candidate will get from 45% to 65% of the votes…

Testing whether samples come from one population

We may also ask whether the samples come from the same population. Let's imagine that we got exit-poll results from 3 more places and we are wondering whether the structure of support is the same in all the places. Say, we got the following result: * Site 1 – 55 out of 100 declared support. * Site 2 – 75 out of 150 declared support. * Site 3 – 455 out of 1000 declared support. * Site 4 – 45 out of 87 declared support. We use the same test for proportions as before:

```
votes <- c(55, 75, 455, 45)
votes.tot <- c(100, 150, 1000, 87)
prop.test(votes, votes.tot)
```

```
##
##  4-sample test for equality of
##  proportions without continuity
```

```
##  correction
##
## data:  votes out of votes.tot
## X-squared = 4.7848, df = 3, p-value =
## 0.1883
## alternative hypothesis: two.sided
## sample estimates:
##    prop 1    prop 2    prop 3    prop 4
## 0.5500000 0.5000000 0.4550000 0.5172414
```

It seems the results come from the same population.

Goodness-of-fit test

Statistical tests can be very useful when visiting Las Vegas, Monte Carlo or when simply playing dice with a stranger. We may easily detect if someone is cheating. Consider a series of results coming from throwing a single dice. We got: * one - 7 times, * two - 14 times, * three - 9 times, * four - 11 times, * five - 15 times, * six - 5 times,

Now, we are wondering whether the dice is fair…

```
results <- c(7,14,9,11,15,5)
probs <- rep(1/6, 6)
chisq.test(x=results, p=probs)
```

```
##
##  Chi-squared test for given
##  probabilities
##
## data:  results
## X-squared = 7.5574, df = 5, p-value =
## 0.1824
```

Well, it seems we can safely continue playing. *Alea iacta est…*} Here we performed a goodness-of-fit test, checking whether the distribution of our results fits the uniform distribution of $\frac{1}{6}$ for each outcome that we expect for a fair dice.

χ^2 *test for independence*

Say that we are wondering whether a new drug has any effect. We can administer drug to a group of N randomly selected patients and administer *placebo* to the same N number of randomly chosen patients. Now, imagine that there three possible outcomes observable after some time of treatment: * patient got better * no change * patient got worse We can use the χ^2 test for independence to see whether treatment has any effect:

```
# Treated group
group1 <- c(21, 39, 40)
group2 <- c(2, 45, 53)
data <- data.frame(group1, group2)
chisq.test(data)
```

As we can see, we can reject H_0 : *group*1 *and group*2 *are not independent*! This means there is a difference between the "treated" and the *placebo* group.

Restoring normality

Sometimes you can restore normality of your variable x by using one of the transformations specified in this document.

References