Lecture 8: Non-parametric Statistics & Simulation

Thomas Chadefaux

Quantitative Methods I

Today

1. Non-parametric tests
2. Bootstrapping

Non-parametric tests

Non-parametric tests: introduction

- Methods of inference typically assume normal distribution in the population

But wait, you said that what is assumed is the normality of the test statistic, NOT the distribution of the population. Because of the central limit theorem, assuming the normality of the test statistic was ok

Non-parametric tests: introduction

- True: Technically what is assumed is normality of the sample means.
- **But** when n is small, or the distribution of x is too extreme, the Central Limit Theorem does not apply.
- In that case, we need to assume that the population itself is distributed normally.

Non-parametric tests: introduction

- In practice, never perfectly normal, but our methods are robust. They can handle some non-normality
- But sometimes non-normality is a problem. In particular when:
  - Too 'non-normal'
  - Outliers
  - Small sample

What to do when normality is violated?

Basic options:

1. Remove the outliers
   - IFF the outliers are due to, say, bad measurement.
   - But often the outliers are reasonable and valuable info., and you should NOT remove them.
2. Transform your data (e.g., log)
3. Rely on other distributions (e.g., Weibull for survival analysis—not covered here)
4. Bootstrap methods and related (later in this lecture)
5. Nonparametric methods that do not require any specific form for the distribution of the population. Common nonparametric methods do not make use of the actual values of the observations, but instead use counts/ranks of observations.

Comparison of tests based on normal distributions with non-parametric tests

Setting	Normal test	Rank test
One sample	One-sample t-test	Wilcoxon signed rank test
Matched pairs	Paired student t-test	
Two Independent samples	independent-samples t test	Wilcoxon rank sum test
Several independent samples	F-test (not covered here)	Kruskal-Wallis test (not covered here).

The Wilcoxon signed-rank test

The Wilcoxon signed-rank test

- Alternative to t-test
- Used to determine whether the **median** of the sample is equal to a known standard value
- Applies to single samples (i.e., difference from zero)
- null hypothesis: the numbers of differences in each direction are equal.

Single sample: An example

Suppose we have one sample: $\{-1, 2, 3, -4\}$. The signed rank test goes as follows:

1. sort the numbers: $-4, -1, 2, 3$
2. Assign a rank to the numbers: $1, 2, 3, 4$ (Note: exclude numbers $= 0$)
3. Compute the test statistic:

$$W = \sum_i sign(number_i) \times rank_i$$

where $sign(number)$ takes value -1 if the number is negative, 0 if 0, and $+1$ if positive.

Single sample: An example

Here,

$$W = -1 \times 1 + (-1) \times 2 + 1 * 3 + 1 * 4 = 4$$

(Note: R gives a slightly different result because they remove the minimum value. The literature is not unanimous on whether or not it should be removed.)

(Single sample: technicalities)

Under the null hypothesis, W follows a specific distribution with no simple expression.

W can be compared to a critical value from a reference table. The two-sided test consists in rejecting H_0 if $|W| > W_{critical,N_r}$, where N_r is the reduced sample size after we remove the 0s.

(Single sample: technicalities)

As N_r increases, the sampling distribution of W converges to a normal distribution.

Thus, for $N_r \geq 20$, a z-score can be calculated as $z = \frac{W}{\sigma_W}$, where $\sigma_W = \sqrt{\frac{N_r(N_r+1)(2N_r+1)}{6}}$

(Single sample: technicalities)

	alpha values						
n	0.001	0.005	0.01	0.025	0.05	0.10	0.20
5	--	--	--	--	--	0	2
6	--	--	--	--	0	2	3
7	--	--	--	0	2	3	5
8	--	--	0	2	3	5	8
9	--	0	1	3	5	8	10
10	--	1	3	5	8	10	14
11	0	3	5	8	10	13	17
12	1	5	7	10	13	17	21
13	2	7	9	13	17	21	26
14	4	9	12	17	21	25	31
15	6	12	15	20	25	30	36
16	8	15	19	25	29	35	42
17	11	19	23	29	34	41	48
18	14	23	27	34	40	47	55
19	18	27	32	39	46	53	62
20	21	32	37	45	52	60	69
21	25	37	42	51	58	67	77
22	30	42	48	57	65	75	86
23	35	48	54	64	73	83	94
24	40	54	61	72	81	91	104
25	45	60	68	79	89	100	113
26	51	67	75	87	98	110	124
27	57	74	83	96	107	119	134

	alpha values						
n	0.001	0.005	0.01	0.025	0.05	0.10	0.20
28	64	82	91	105	116	130	145
29	71	90	100	114	126	140	157
30	78	98	109	124	137	151	169
31	86	107	118	134	147	163	181
32	94	116	128	144	159	175	194
33	102	126	138	155	170	187	207
34	111	136	148	167	182	200	221
35	120	146	159	178	195	213	235
36	130	157	171	191	208	227	250
37	140	168	182	203	221	241	265
38	150	180	194	216	235	256	281
39	161	192	207	230	249	271	297
40	172	204	220	244	264	286	313
41	183	217	233	258	279	302	330
42	195	230	247	273	294	319	348
43	207	244	261	288	310	336	365
44	220	258	276	303	327	353	384
45	233	272	291	319	343	371	402
46	246	287	307	336	361	389	422
47	260	302	322	353	378	407	441
48	274	318	339	370	396	426	462
49	289	334	355	388	415	446	482
50	304	350	373	406	434	466	503

Two paired samples

Same process, except that instead of raw numbers, we use the difference:

1. Calculate the absolute value of the difference and the sign: $|x_{1,i} - x_{2,i}|$ and $sign(x_{1,i} - x_{2,i})$
2. order and rank the pairs from smallest to largest absolute difference
3. Calculate the test statistic:

$$W = \sum_i sign(x_{2,i} - x_{1,i}) \times rank_i$$

Two paired samples: an example

Suppose we have the following data:

1,2

1,3

4,2

1,5

3, 0

Two paired samples: an example

abs: 1, 2, 2, 4, 3

sign: +1, +1, -1, +1, -1

rank: 1, 2.5, 2,5, 4, 3

so W= $2 + 2.5 - 2.5 + 4 - 3 = 3$

the Wilcoxon Signed-Rank Test: Summary

In short: Use the Wilcoxon Signed-Rank Test to compare the median of a set of numbers to a hypothetical median.

I.e., equivalent to one-sample t-test, but without assumption of normality.

This is esp. useful when you have a very small sample, or your data is particularly non-normal.

The Wilcoxon rank sum test (aka Mann–Whitney U test)

The Wilcoxon rank sum test

- Tests the null hypothesis that a randomly selected value from one sample is equally likely to be less than or greater than a randomly selected value from a second, independent, sample.
- I.e., test the null hypothesis that $P(X > Y) = 0$
- Alternative to the two sample t-test without assuming normal distributions

The Wilcoxon rank sum test

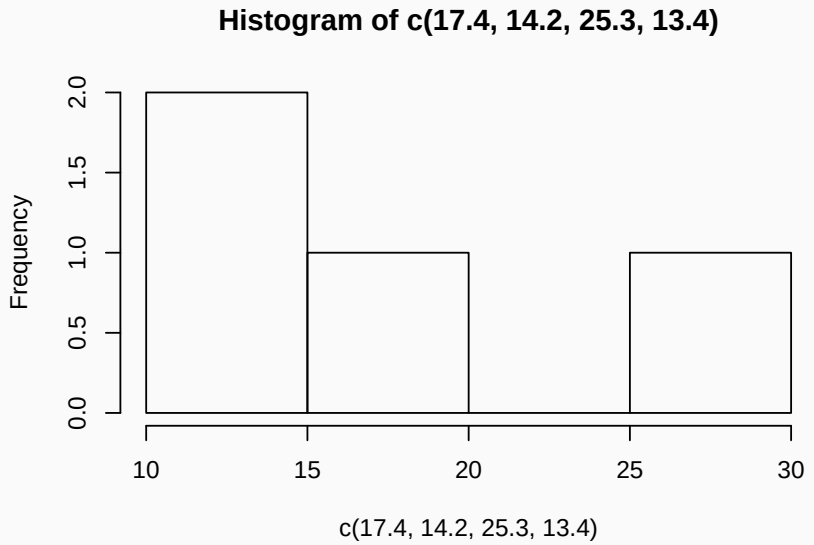
Example:

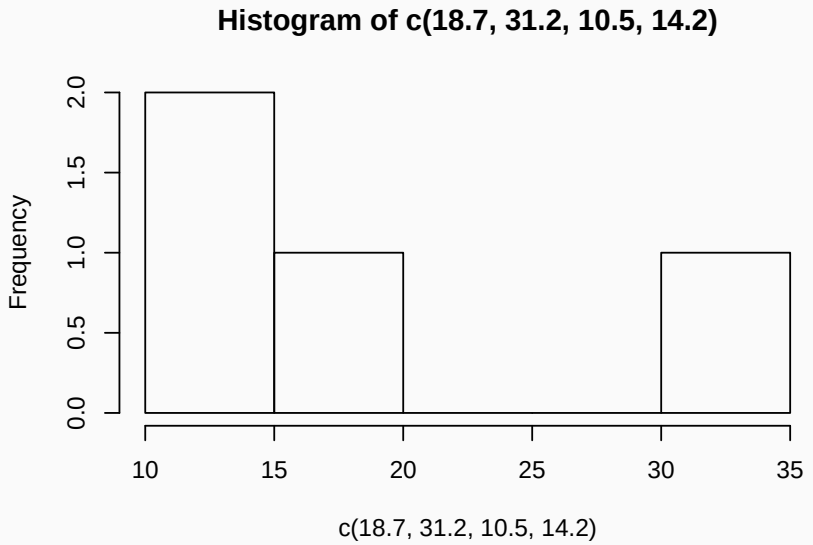
- Suppose we are interested in the effect of TV soap operas on ethnic relations in Burundi.
- We assign a treatment group of 4 individuals to watching TV one hour a day for a month, and a control group of another 4 to watching no TV.
- At the end we ask them to make a donation to a charity that will support the other ethnic group. We record the following donations, in Burundian Francs:

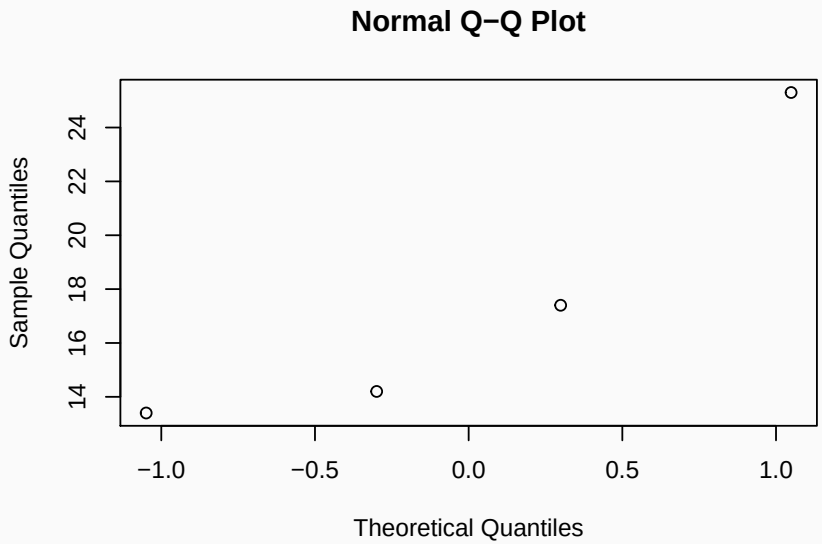
TV?	Donations			
Yes	17.4	14.2	25.3	13.4
No	18.7	31.2	10.5	14.4

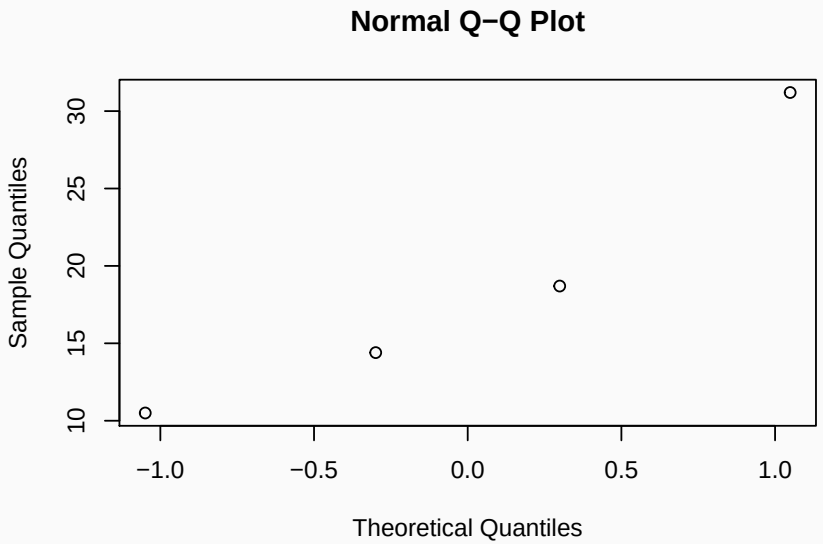
Does TV have a significant impact on donations?

First, is the data normally distributed?









Really hard to tell. . . Sample size is too small.

The Wilcoxon rank sum test: Direct Method

For each obs. in one set:

- Count the number of times this first value wins over any obs. in the other set.
- Count 0.5 for ties.
- The sum of wins and ties is U for the first set.

Method 1: Direct Method: an example

Exposure to TV?	Donations			
Yes	17.4	14.2	25.3	13.4
No	18.7	31.2	10.5	14.4

- 17.4 'wins' against 10.5 and against 14.2, so it wins 2 times.
- 14.2 wins against 10.5, so 1 time
- 25.3 wins against 18.7, 10.5 and 14.4, so 3 times
- 13.4 wins against 10.5, so 1 times

So $W = 2 + 1 + 3 + 1 = 7$. The distribution of W is rather complex, so you'll need the software to check its significance.

Method 1: Direct Method: an example (cont'd)

Check with R:

```
A <- c(17.4, 14.2 , 25.3, 13.4)
B <- c(18.7, 31.2, 10.5, 14.4)
wilcox.test(A,B)
```

```
##
##  Wilcoxon rank sum test
##
## data:  A and B
## W = 7, p-value = 0.8857
## alternative hypothesis: true location shift is not equal
```

The Wilcoxon rank sum test (aka Mann-Whitney U test)

Recap: the Wilcoxon rank sum test is testing the null hypothesis that $P(X > Y) = 0$.

Note that the test is not symmetric:

```
x1 <- c(1,2,5)
x2 <- c(3,4,7)
wilcox.test(x1, x2)
```

```
##
##  Wilcoxon rank sum test
##
## data:  x1 and x2
## W = 2, p-value = 0.4
## alternative hypothesis: true location shift is not equal
```

```
wilcox.test(x2, x1)
```

In-class exercise with R (1)

Consider the following two samples. Calculate W (aka U)

$$A = \{34, 1, 91, 88, 43\}$$

$$B = \{39, 98, 76, 15, 51\}$$

1. Do it manually

In-class exercise

Consider the following two samples. Calculate W (aka U)

$$A = \{34, 1, 91, 88, 43\}$$

$$B = \{39, 98, 76, 15, 51\}$$

1. Do it manually

$$U_A = 1 + 0 + 4 + 4 + 2 = 11$$

$$U_B = 2 + 5 + 3 + 1 + 3 = 14$$

In-class exercise

2. Write a function in R that will return W for any two samples

In-class exercise

```
myman.u <- function(A, B){
      U <- 0
      for(i in 1:length(A)){
            U <- U + length(which(A[i] > B))
            }
      return(U)
      }
```

Let's test our function

```r
# test with new samples
A <- rnorm(100); B <- rnorm(100)
myman.u(A,B)
```

```
## [1] 5732
```

```r
#Verify using R's canned function
myman.u(A,B) == wilcox.test(A,B)$statistic
```

```
##    W
## TRUE
```

Bootstrapping

The Bootstrap

- Bootstrap: "loop sewn at the top rear or sometimes on each side of a boot to facilitate pulling it on."
- Baron von Muenchhausen pulled himself (and his horse) up by his own bootstraps...

Bootstrapping

- Fundamental idea of bootstrapping: use computing power to ask: what would happen if we repeated this method many times?
- The bootstrap is a way of finding the sampling distribution, at least approximately, from just one sample
- Basic idea: Sample from your data, with sample size n. Compute your statistic. Repeat many times to obtain the distribution of that statistic.
- From that distribution, you can obtain the mean, SE, CI, etc.

Why bootstrap?

- Issues with normality assumption
- Small sample
- Theoretical distribution of a statistic of interest is complicated or unknown
  - What if you are interested in, say, the *ratio* of means? Or the confidence interval of the median?
  - No simple traditional method for it
- Simpler!
- Appeals directly to the basis of all inference: the sampling distribution that shows what would happen if we took very many samples under the same conditions
- In many fields the preferred way to do inference (sadly less so in the social sciences, but still frequent)

Resampling

1. Resample
    - Typically we only have one sample. The idea here is to create many resamples by sampling *with replacement* from the population
        - 'With replacement' means that we put the observation back before drawing the next. An obs. can therefore be drawn more than once
2. Calculate the statistic for each resample
3. Calculate the statistic of these statistics. E.g.,
    - mean
    - CI

Resampling

The original sample represents the population from which it was drawn. Thus, resamples from this original sample represent what we would get if we took many samples from the population.

The bootstrap distribution of a statistic, based on the resamples, represents the sampling distribution of the statistic

Bootstrapping: Example

- Suppose 4,000 students are supportive of the CETA (Canada-EU trade agreement), whereas 8,000 are opposed
- We want to find the CI of the mean.
- Ideally we use the exact method we learned in the last weeks, but maybe we have a doubt–are the assumptions satisfied, etc.
- So, bootstrap!

Bootstrapping: Example

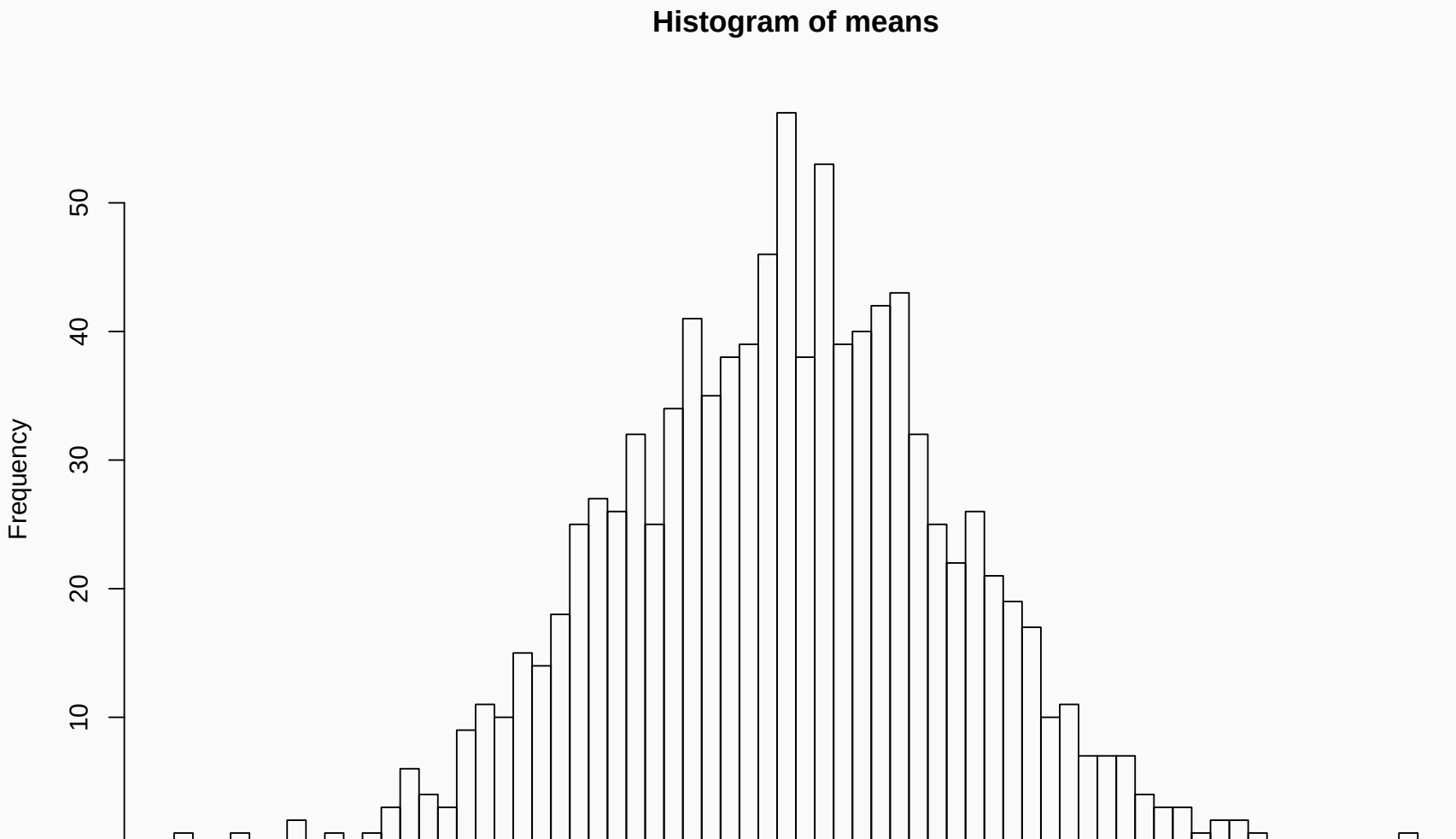
```
#Simulate fake data:
CETA <- c( rep(1, 4000), rep(0, 8000))
# bootstrap and calculate means
means <- replicate(n = 1000,
      expr = mean(sample(CETA,
                         size = 12000,
                         replace=T)))
```

`replicate` asks R to repeat a particular expression n times (here 1000). So here we are asking R to collect 1000 samples of size 12000, drawn from "population" (not really a population of course) CETA, with replacement.

Bootstrapping: Example

Now we have 1000 samples, and we can see, for example, how they are distributed:

```
hist(means, breaks=50)
```



Bootstrapping: Example

This gives us our bootstrapped sampling distribution! Now we can just pluck values such as the CI:

```
sort(means)[c(25,975)]
```

```
## [1] 0.3248333 0.3419167
```

Not exactly the same as our "proper" result of (0.3249, 0.3418), but pretty close.

Bootstrapping: Example

In fact, increasing the number of resamples gets us very close:

```
means <- replicate(n = 10000,
                   expr = mean(sample(CETA, size = 12000, r
sort(means)[c(250,9750)]
```

```
## [1] 0.3249167 0.3419167
```

Bootstrapping: Example

We can also use this bootstrapped sampling distribution to estimate the standard error:

```
sd(means)
```

```
## [1] 0.004312956
```

Note that the estimate of the SE is also very close to the value estimated using the formula, i.e., 0.00430

Standard error of the median

Suppose now you wanted to calculate the standard error of the median. It's very simple:

Standard error of the median (ignore this slide and the next!)

Suppose now you wanted to calculate the standard error of the median. It's very simple:

$$f_m(x) = g(c(x))f(x),$$

where

$$g(x) = \frac{(1-x)^{\frac{n-1}{2}} x^{\frac{n-1}{2}}}{B\left(\frac{n+1}{2}, \frac{n+1}{2}\right)},$$

where B is the beta function, $c(x)$ is the cumulative distribution function of the sample distribution, and $f(x)$ is the probability density function of the sample distribution.

Standard error of the median

Now the expected value of the sample median is:

$$\mu_m = \int x f_m(x) dx$$

and the standard deviation of the sample median is:

$$\sigma_m = \sqrt{\int (x - \mu_m)^2 f_m(x)\ dx}$$

See, simple!

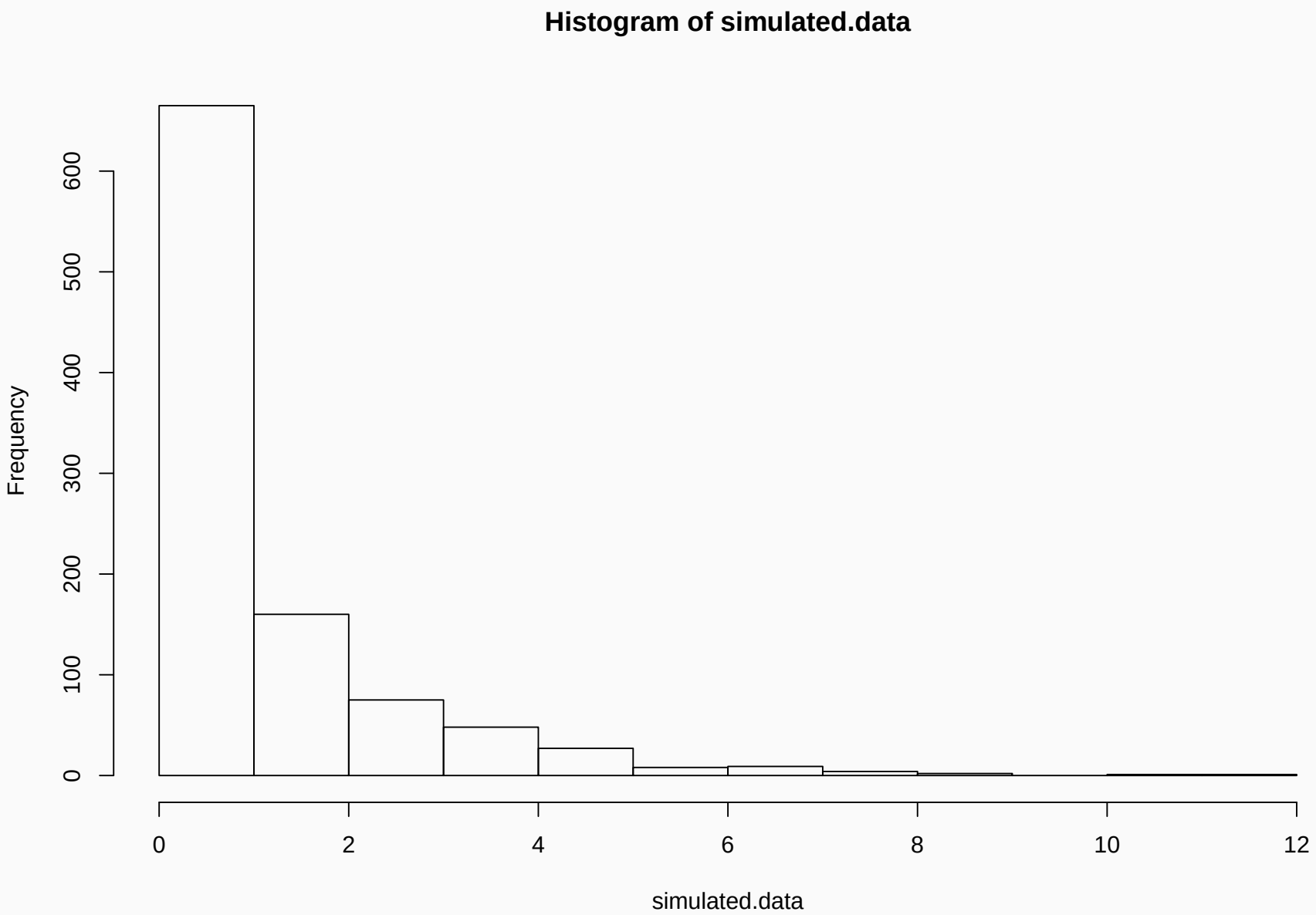
Standard error of the median

Can we do the same thing using bootstrapping? Let's generate random data from, say, a chi-square dist.:

```
n <- 1000
simulated.data <- rchisq(n, df = 1)
```

Standard error of the median

```
hist(simulated.data)
```



In-class exercise: calculate the confidence interval of the median

```
medians <- replicate(n = 1000,
                     expr = median(sample(simulated.data,
                                          size = n,
                                          replace=T)))
sort(medians)[c(25,975)]
```

```
## [1] 0.3994001 0.5162531
```

and you're done!

Why not ALWAYS use the bootstrap?

If your data has rare extreme values, bootstrapping might never pick these observations, or "under" pick them

$\rightarrow$ It might underestimate the variability in the underlying population.

In practice: why not use both bootstrap **and** standard techniques?

Bootstrap the US election!

Simulate pre-election polls—1 for Biden, 0 for Trump

- Ohio: 50.5% Biden, 49.5% Trump
- Pennsylvania: 52% Biden, 48% Trump
- Florida: 49.9% Biden, 50.1% Trump

For the sake of the argument, suppose that these are the only three states, and that each state has 1 electoral college (so you need 2 to win).

In-class exercise: Estimate Biden's probability of winning the election

Bootstrap the US election!

Estimate Biden's probability of winning

1. Set up the fake data (note: could be any other large numbers that matches the % above:

```
nobs <- 10000
Ohio <- c(rep(1, nobs*0.505), rep(0, nobs*0.495))
Penn <- c(rep(1, nobs*0.52), rep(0, nobs*0.48))
Florida <- c(rep(1, nobs*0.499), rep(0, nobs*0.501))
```

Bootstrap the US election!

2. Estimate p(winning) in each state

```
nrep <- 10000
means.ohio <- replicate(n = nrep,
                        mean(sample(Ohio, n, replace=T)))
means.penn <- replicate(n = nrep,
                        mean(sample(Penn, n, replace=T)))
means.florida <- replicate(n = nrep,
                        mean(sample(Florida, n, replace=T)))
```

Bootstrap the US election!

3. Given Biden's score in each state, how often does he get two electors?

```
electors <- cbind(round(means.ohio),
                  round(means.penn),
                  round(means.florida))
wins <- rowSums(electors)
length(wins[wins >= 2]) / length(wins)
```

```
## [1] 0.7368
```

Bootstrap the US election!

Congratulations, you have passed level 1

Bootstrap the US election!

Level 2: What is your confidence interval around your estimate of Biden's probability of winning?

An aside: Monte-Carlo simulation

- Goal is to test drive estimators
- Generate data that depends on some parameters.
- Boostrapping is based on unknown distribution, and Monte Carlo based on known distribution.
- Both bootstrapping and Monte Carlo simulation use repetitive sampling and then examine the results.
- But whereas bootstrapping uses the initial sample as the population from which to resample, Monte Carlo simulation is based on setting up a data generation process (with known values of the parameters).