Statistical Methods for High Dimensional Biology

STAT/BIOF/GSAT 540

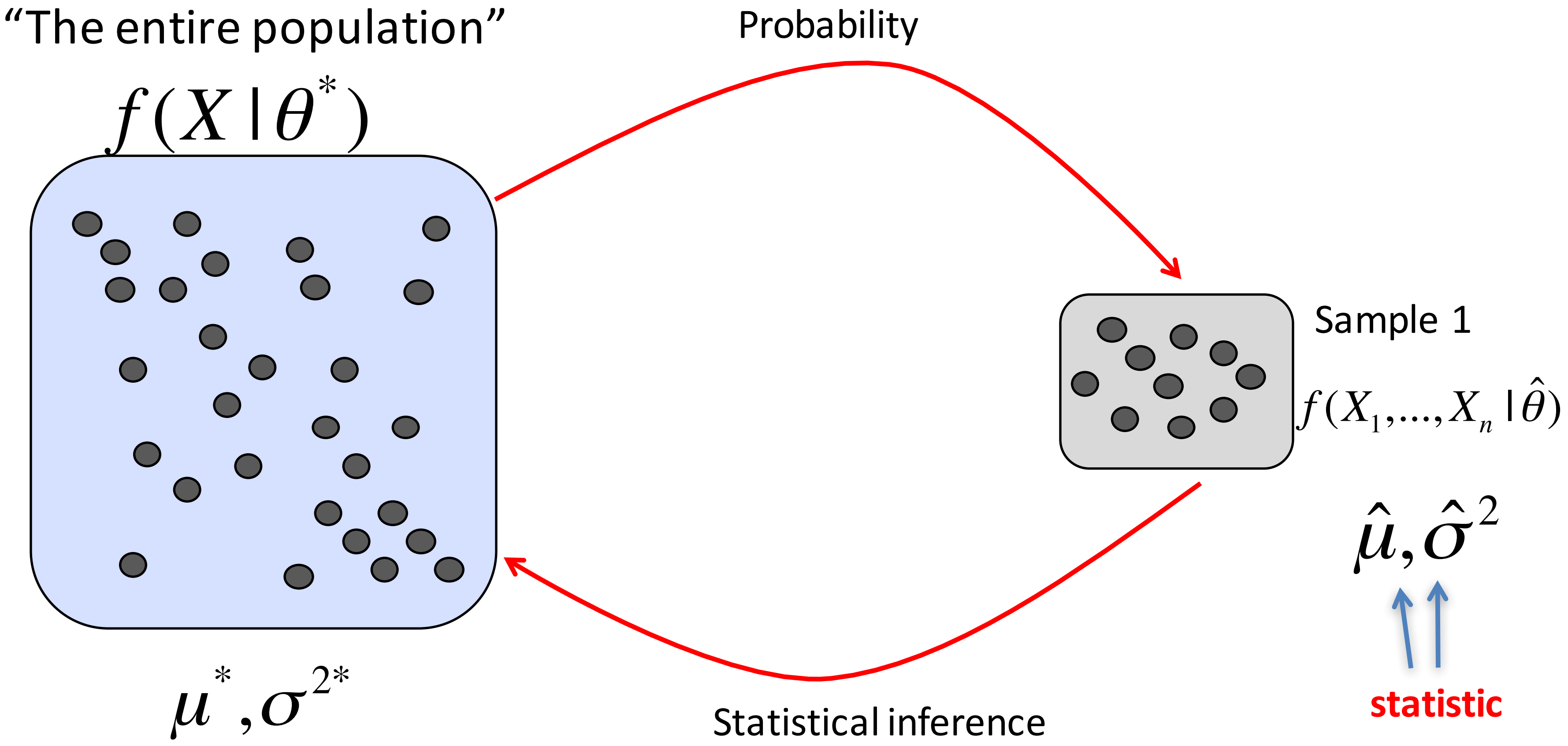
Lecture 20 – Bootstrap & Permutation Testing

Amrit Singh

March 22 2017

Based on slides by Drs. Mostafavi & Bryan

Central Dogma of Statistics



“The entire population”

$f(X|\theta^*)$

Probability

Sample 1

$f(X_1,\dots,X_n|\hat{\theta})$

$\hat{\mu},\hat{\sigma}^2$

statistic

μ^*,σ^{2*}

Statistical inference

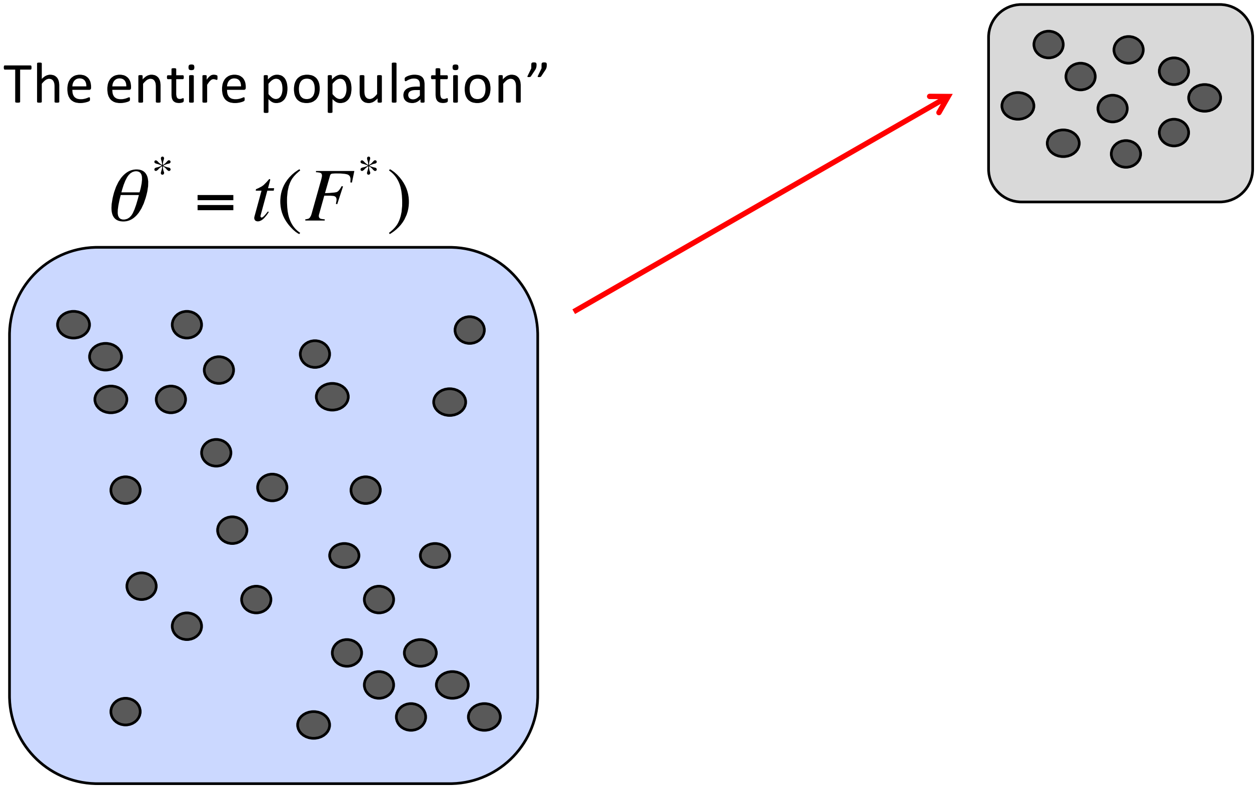
Statistical Inference

- We are given a sample (i.e., some data) $X_1,\dots,X_n$ that are independent draws from an underlying data generating function $f(X \mid \theta^*)$

- We want to known something about *f*, for example we want to know θ^*

- An estimate $\hat{\theta}$ is just some function of $X_1,\dots,X_n$, for example you can think of it as $\hat{\theta} = \hat{\theta}(X_1,\dots.,X_n)$

- If we could repeat our "experiment", we could get sampling distribution for $\hat{\theta}$



“The entire population”

$$\theta^{*} = t(F^{*})$$

Sample 1

estimate

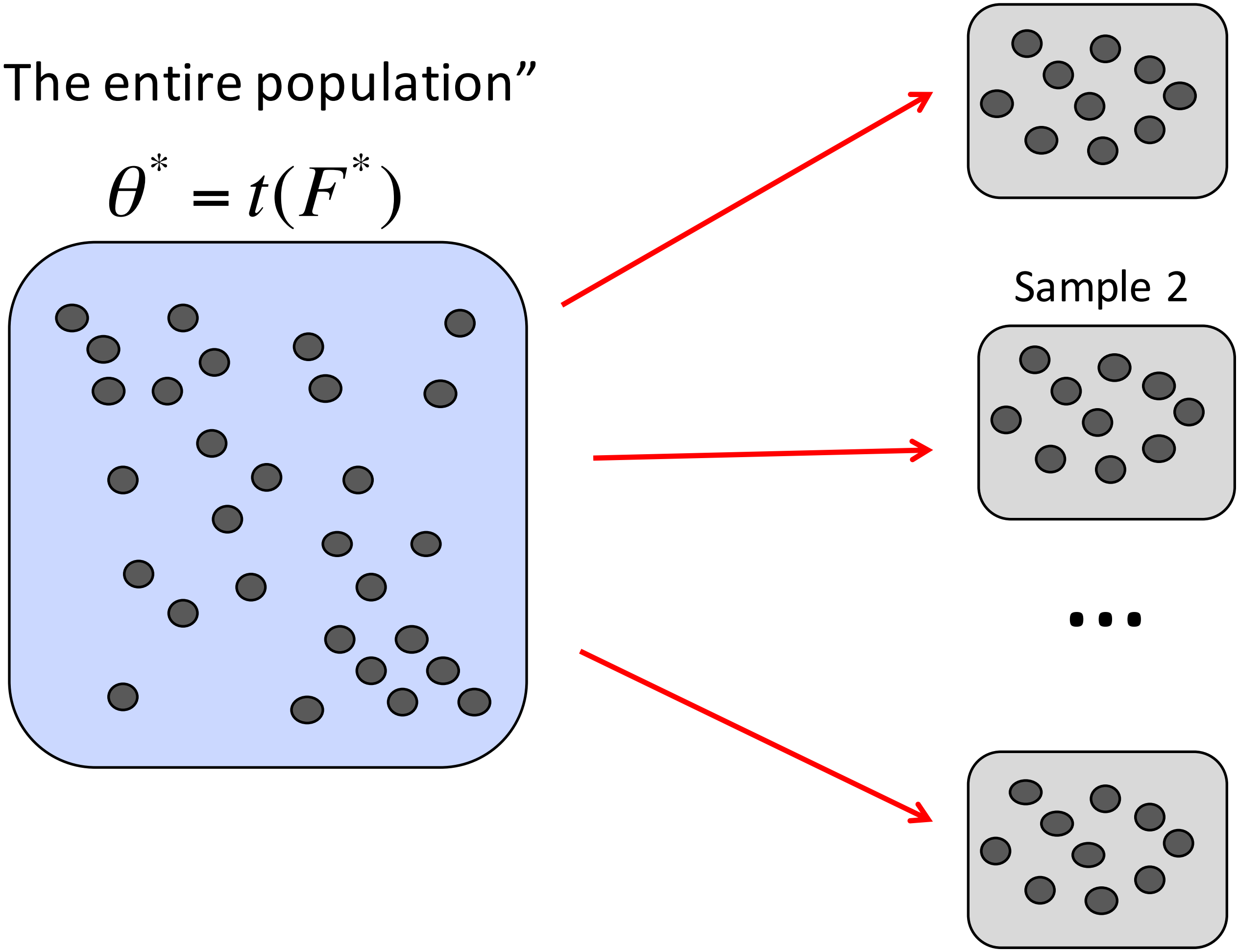
$$\hat{\theta}_1 = t(\hat{F}_1)$$

e.g.

$$\hat{\mu}_1$$

“The entire population”

$$\theta^* = t(F^*)$$



Sample 1

estimate

e.g.

$$\hat{\theta}_1 = t(\hat{F}_1)$$

$$\hat{\mu}_1$$

Sample 2

$$\hat{\theta}_2 = t(\hat{F}_n^2)$$

$$\hat{\mu}_2$$

…

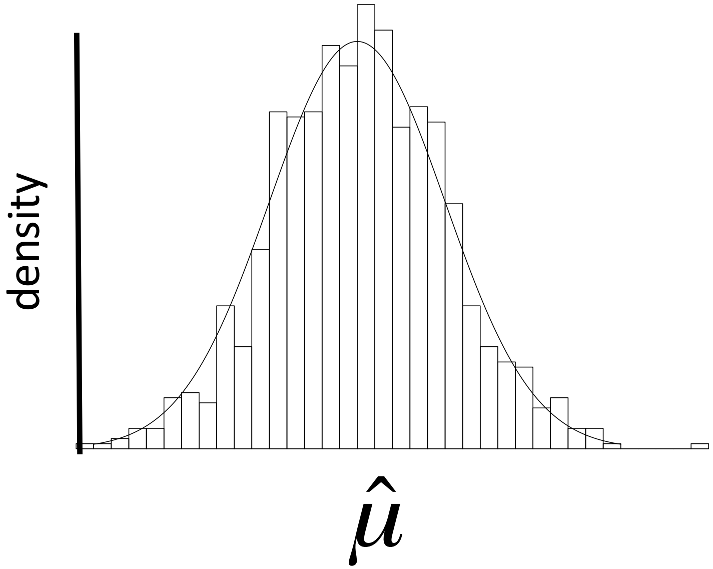
Sample k

$$\hat{\theta}_k = t(\hat{F}_n^k)$$

$$\hat{\mu}_k$$

Sampling distribution of a statistics

- The distribution of the estimates computed from repeating the experiment multiple times: sampling distribution.



- If we had it, we could assess some properties of our estimate:
  - Standard deviation of the estimate ("standard error")
  - Confidence intervals

Sampling distribution

- The sampling distribution, at the moment, is a theoretical construction—it consists of all possible outcomes for experiments that we could have run.

- In practice, we only ran a single experiment to get all our data. But we still want to assess the properties of our estimate. How?

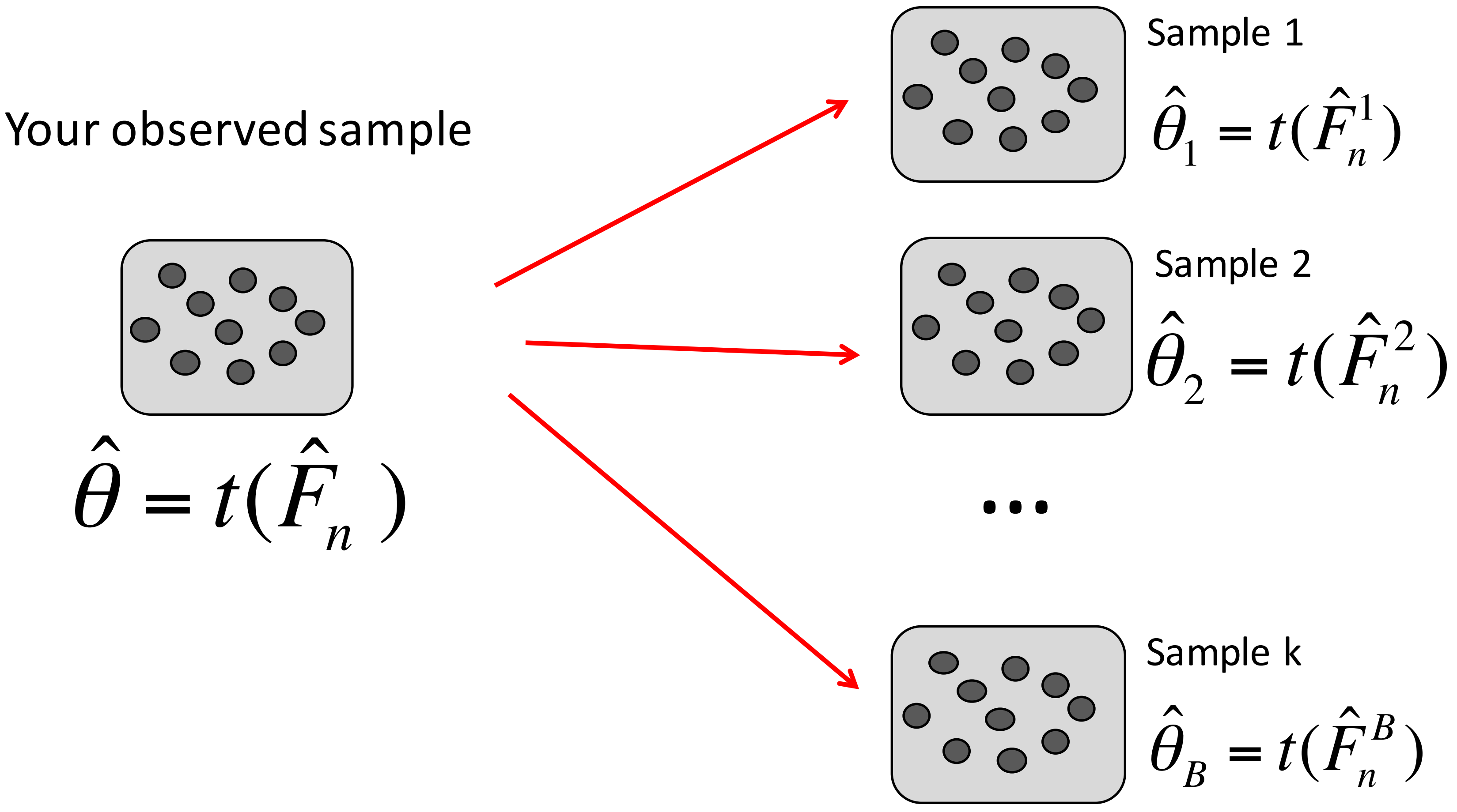
  - Asymptotic theory (aka large sample theory):
    - statistical framework for assessing and modeling the sampling distribution of a statistic
  - Resampling: Bootstrap

Resampling methods

- Ways of performing statistical inference, and quantify uncertainty in our estimates, that are “internal to the data” under analysis: e.g., you get the necessary knowledge about sampling variability (of parameters/estimates) from the observed data itself.

- Resampling methods:
  - Bootstrap: confidence intervals; standard errors; null distribution/hypothesis testing
  - Permutation testing: the null distribution/hypothesis testing
  - Cross-validation: generalization error; setting parameters

The Bootstrap: sample with replacement from the observed data to get sampling distribution for your estimate



Your observed sample

$$\hat{\theta} = t(\hat{F}_n)$$

Sample 1

$$\hat{\theta}_1 = t(\hat{F}_n^1)$$

Sample 2

$$\hat{\theta}_2 = t(\hat{F}_n^2)$$

…

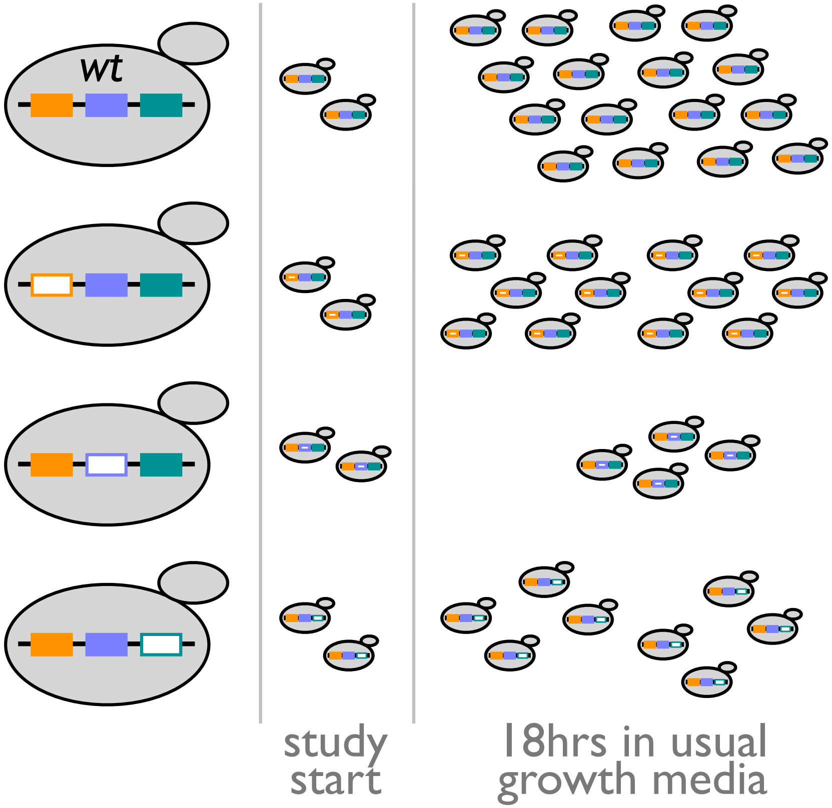
Sample k

$$\hat{\theta}_B = t(\hat{F}_n^B)$$

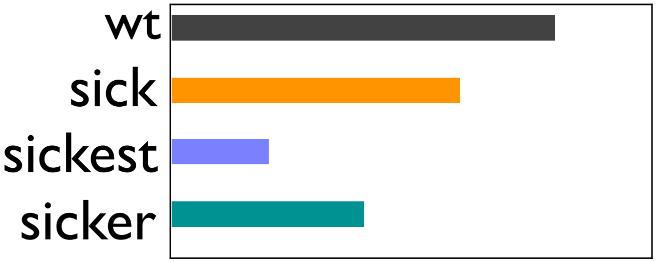
Bootstrap:

Repeat experiment B times (in the bootstrap world) to form b bootstrap replicates of your experiment, then use the B bootstraps to obtain a sampling distribution for your parameter.

Example application of the bootstrap



Study of fitness of
yeast deletion
mutants



Data source: Giaever G, Flaherty P, Kumm J, Proctor M, Nislow C, et al. (2004) Chemogenomic profiling: identifying the functional interactions of small molecules in yeast. Proc Natl Acad Sci U S A 101: 793-798. Pubmed. DOI: 10.1073/pnas.0307490100

Rationale for growth studies of yeast deletion mutants

- Analogy: flipping circuit breakers in a house to determine which lights and outlets are controlled by each circuit

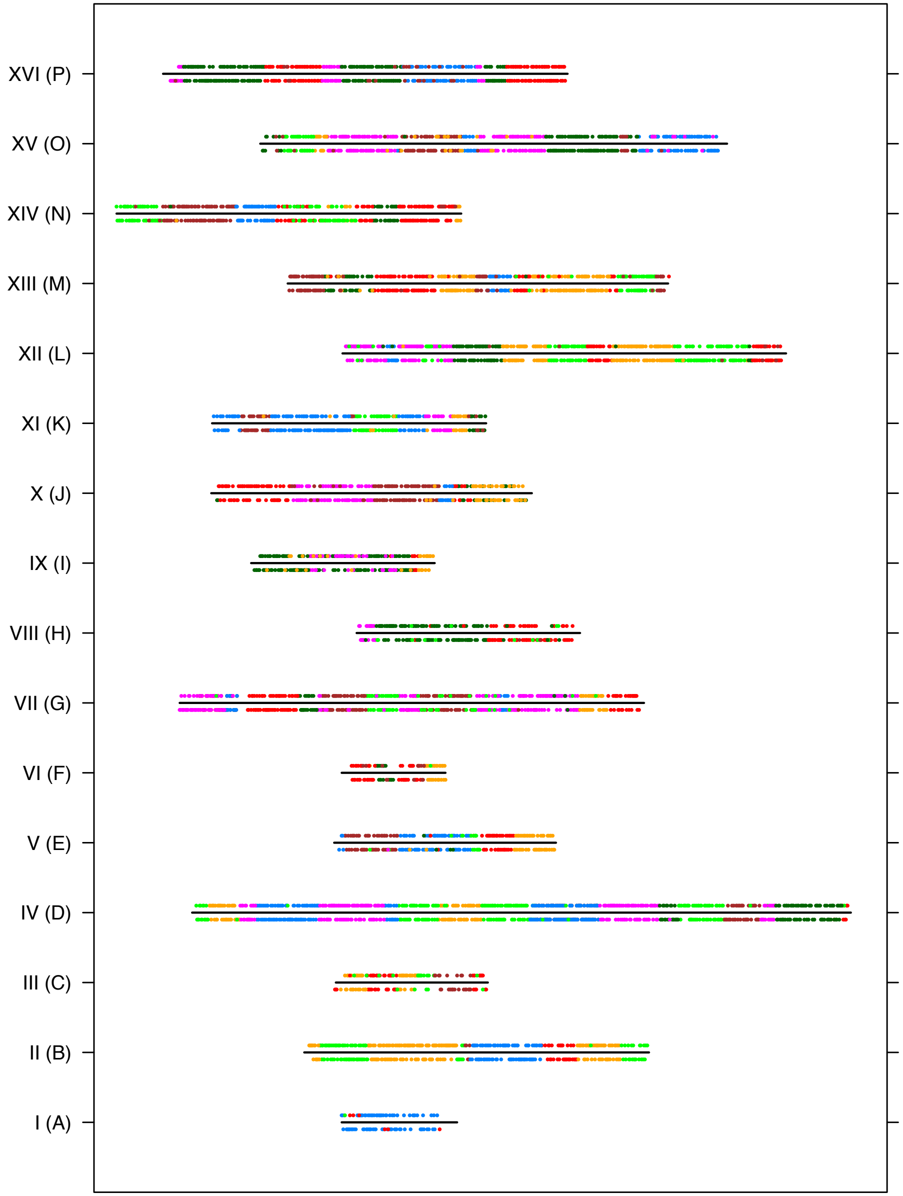
- If the deletion mutant for gene *g* is defective at some biological activity, that suggests that gene *g* contributes to that activity.

- Growth studies are the ‘entry-level’ study. In real life, we often measure more complicated phenotypes and subject the mutant to additional challenges, e.g. treatment with drugs or deletion/mutation of additional genes. Also, this type of data is often integrated with from other types of studies.

Yeast genome has 16 chromosomes.

Each gene lives somewhere on one of these chromosomes.

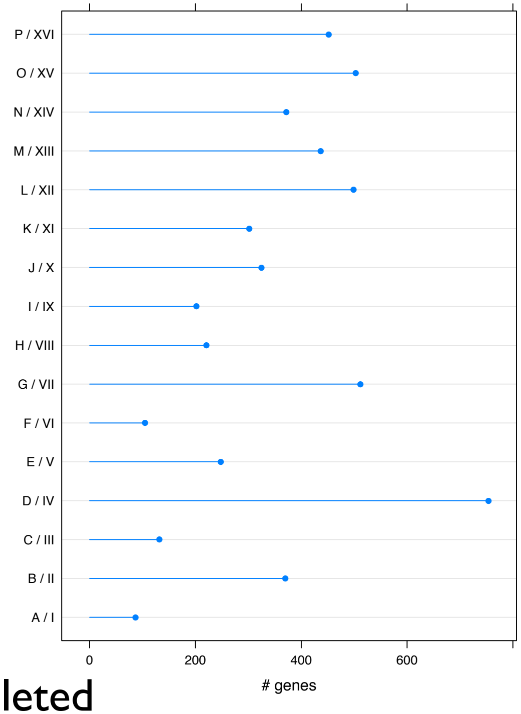
Therefore, each deletion mutant is also associated with one yeast chromosome.

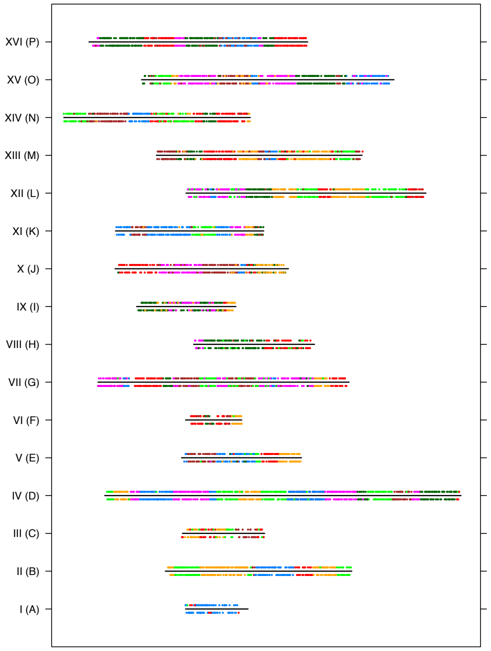


```
> str(hDat)
'data.frame':  5521 obs. of  4 variables:
 $ geneDel     : Factor w/ 5521 levels "YAL001C","YAL002W",..: 1 2 3 4 5 6 7 8..
 $ chromo      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ chromoPretty: Factor w/ 16 levels "A / I","B / II",..: 1 1 1 1 1 1 1 1 1 1 ..
 $ pheno       : num  9.39 9.4 10.38 10.54 8.65 ...
```

```
> peek(hDat)
     geneDel chromo chromoPretty    pheno
190  YBL102W      2       B / II 9.285750
917  YDR089W      4       D / IV 9.528659
1040 YDR185C      4       D / IV 7.079669
1969 YGL201C      7     G / VII 9.754082
2118 YGR046W      7     G / VII 9.262812
3175 YKL085W     11      K / XI 9.479903
3622 YLR176C     12     L / XII 7.359638
```

```
> dotplot(table(hDat$chromoPretty),
+         origin = 0, type = c("p", "h"),
+         xlab = "# genes")
```





Each row consists of

- geneDel = name of the gene that was deleted
- chromo = the associated chromosome (an integer between 1 and 16)
- chromoPretty = a prettier version of the chromosome (more suitable for labeling in tables and figures)
- pheno = a growth phenotype (due to experimental realities and pre-processing, the units are meaningless, i.e. don’t expect to see a cell count here)

Data for our analysis

response = a quantitative measure of growth

e.g. growth rate or # cells at study end

also know the specific yeast gene that was deleted

e.g. YDL133WY = a yeast ORF

and the chromosome on which the gene is found

e.g. "chromosome 4 / D"

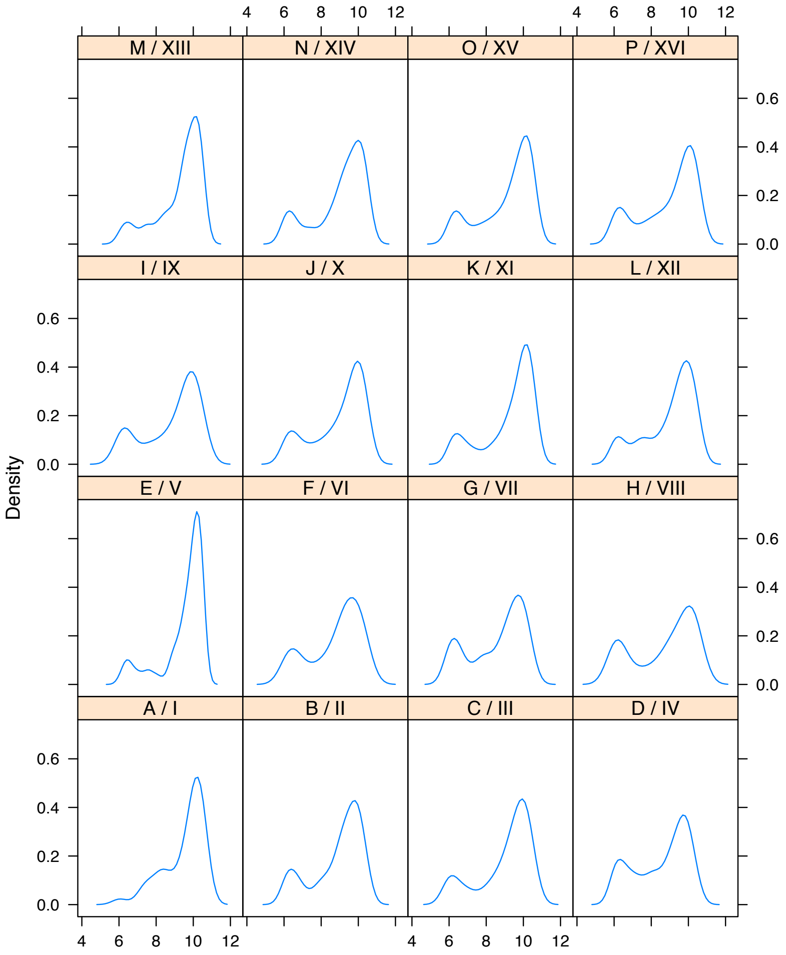
Typical application of bootstrap

Bootstrap

- Estimating key features of the sampling distribution:
  - The standard deviation of the statistics (“standard error”)
  - Confidence intervals
  - Assess whether the asymptotic distribution has started to “kick-in”
  - The bias of an estimate
- Hypothesis testing: constructing the null distribution

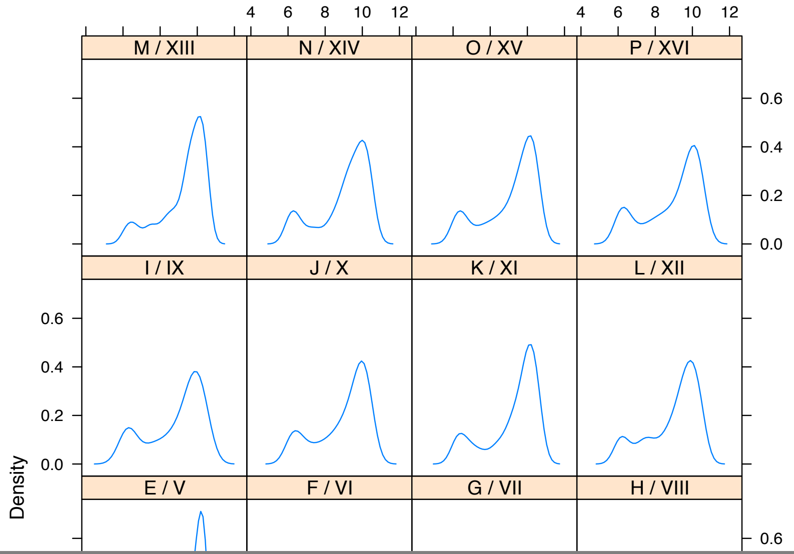
quantitative growth phenotypes for gene deletion mutants

each panel = phenotypes for mutants lacking genes on that chromosome



Data source: Giaever G, Flaherty P, Kumm J, Proctor M, Nislow C, et al. (2004) Chemogenomic profiling: identifying the functional interactions of small molecules in yeast. Proc Natl Acad Sci U S A 101: 793-798. Pubmed. DOI: 10.1073/pnas.0307490100 quantitative growth phenotypes for gene deletion mutants

each panel =

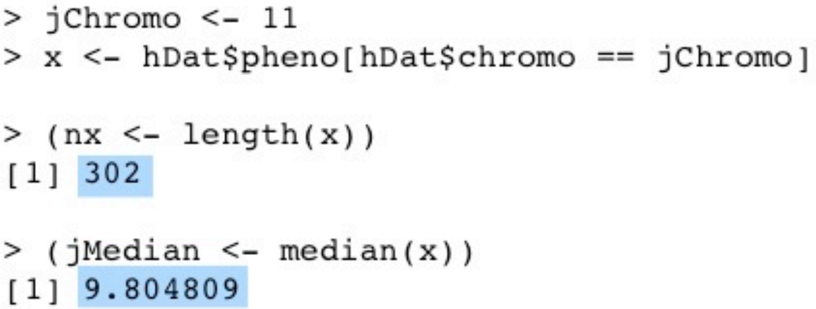


We will use bootstrap to:

a) Assess sampling distribution of the median growth phenotype for a chromosome.
b) Revisit two-group comparison.

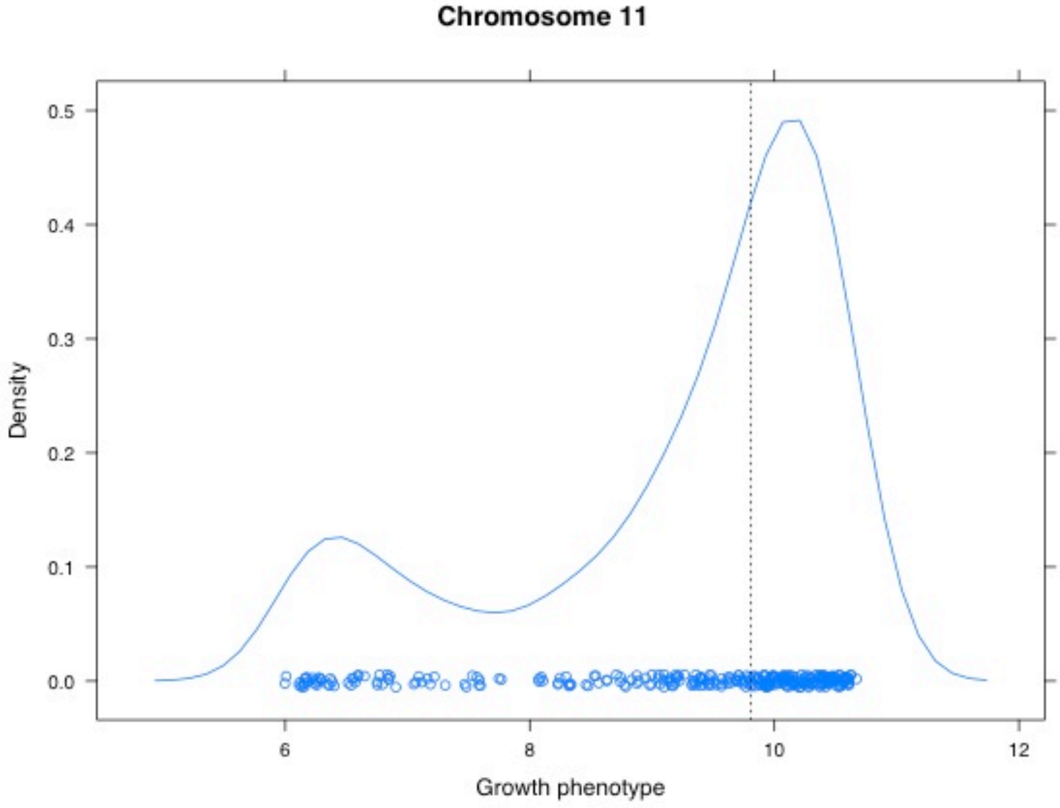
C, et al. (2004) Chemogenomic profiling: identifying the functional interactions of small molecules in yeast. Proc Natl Acad Sci U S A 101: 793-798. Pubmed. DOI: 10.1073/pnas.0307490100

Example: Median for chromosome 11



```
> jChromo <- 11
> x <- hDat$pheno[hDat$chromo == jChromo]

> (nx <- length(x))
[1] 302

> (jMedian <- median(x))
[1] 9.804809
```

- 302 genes on chromosome 11
- Median fitness value if 9.804809



Example: Median of chromosome II

- Large-sample theory says that the sample median is asymptotically **normal** with mean = true median* and variance = 1/4n $f(m)^2$

- Good news asymptotic distribution of median is known.

- Let’s compare this theoretical result to the bootstrap result.

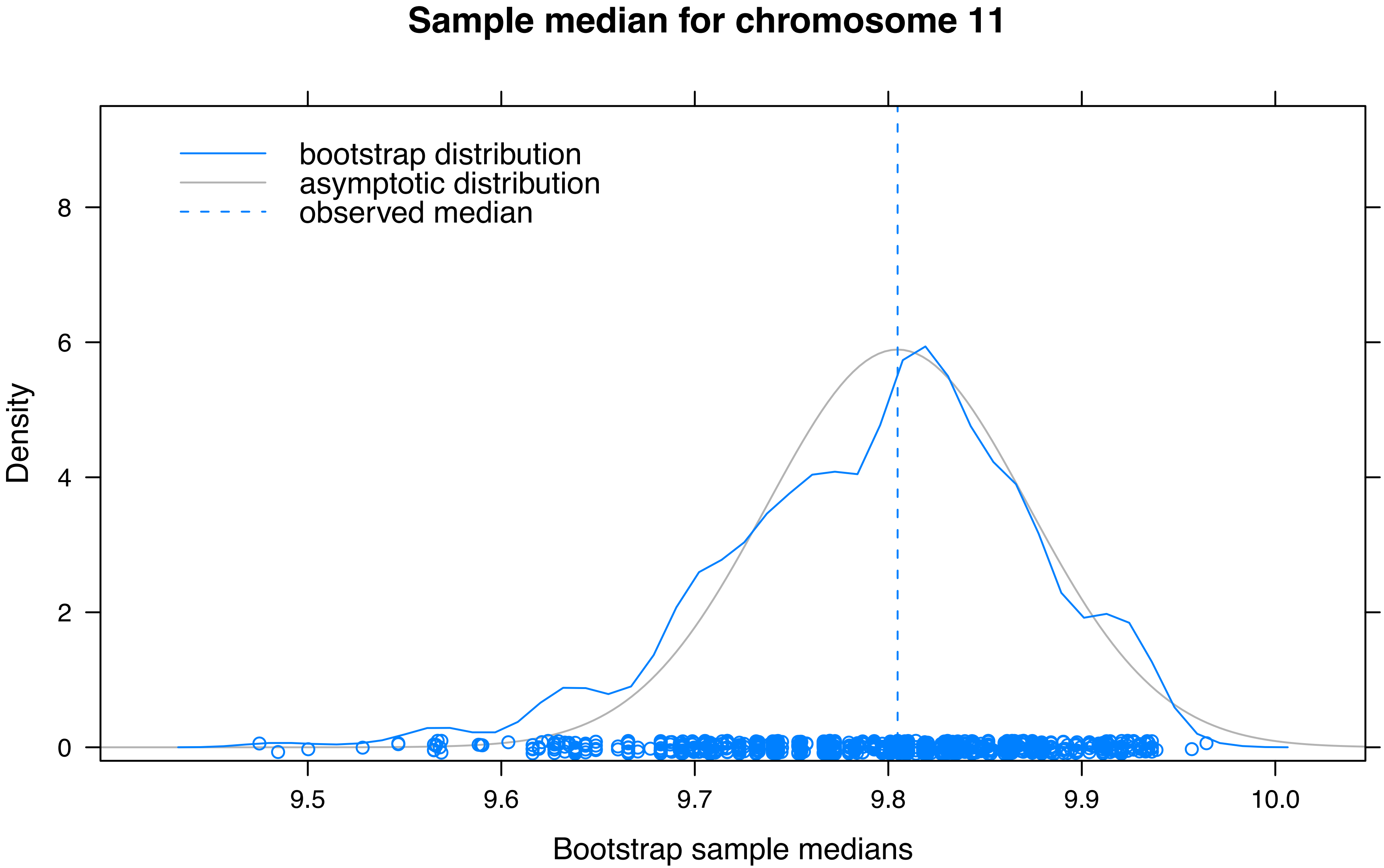
Generating the bootstrap results

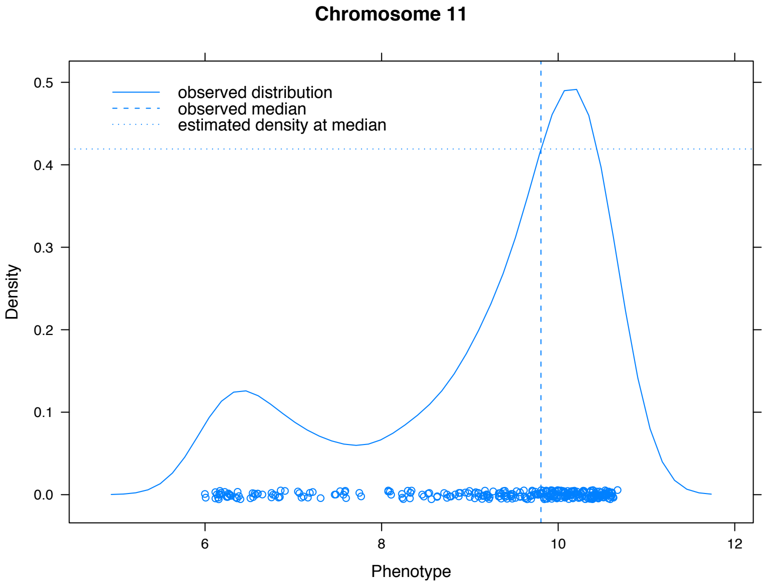
- Draw a new sample of size n=302 from the observed data (response), with replacement, from chromosome II

- Note that we are doing sampling with replacement: this means some observations will re-appear (some once, some twice, etc) and some observations may not appear at al in a given bootstrap sample.

- After each bootstrapping experiment, we take the median of the bootstrap sample. That is the bootstrap statistic.

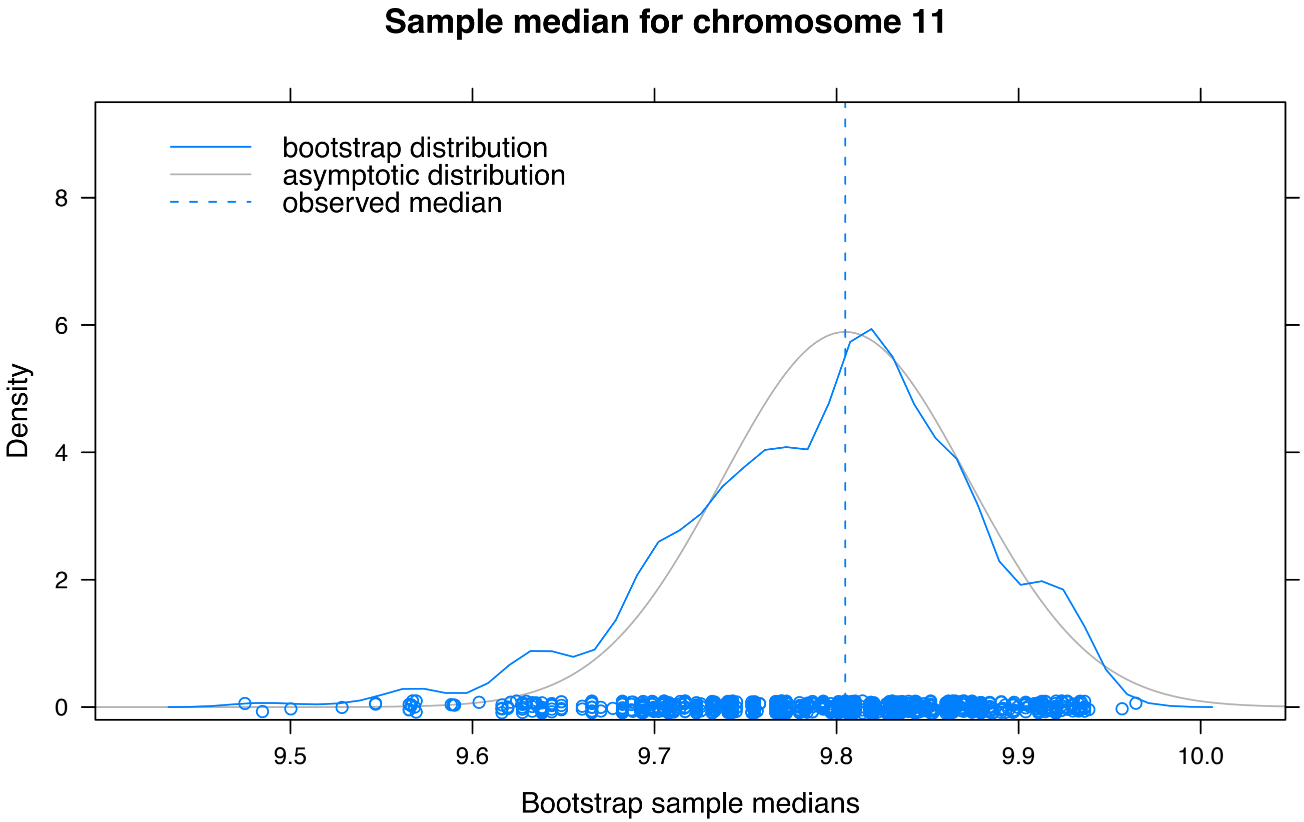
- We do that B times (B should be *large*). We look at at distribution of our bootstrap statistics.

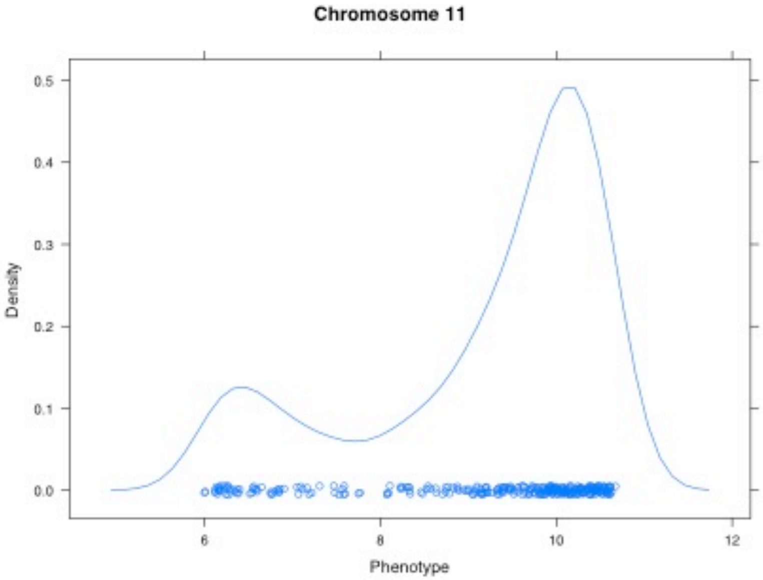

Sample median for chromosome 11
bootstrap distribution
asymptotic distribution
observed median
Density
0
2
4
6
8
9.5
9.6
9.7
9.8
9.9
10.0
Bootstrap sample medians

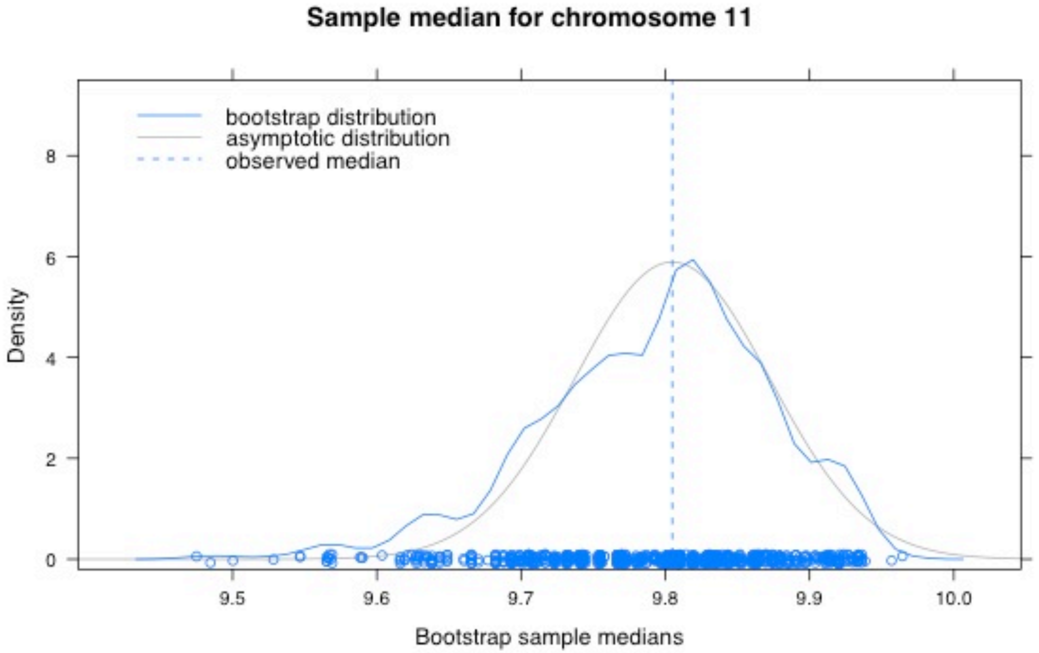


Features we could foresee:

- Both dist'ns have mode @ sample median = 9.8
- Left tail of bootstrap distribution heavier than than that of asymp. norm







```
> B <- 1000

> bootData <-
+   matrix(sample(x, size = B * nx, replace = TRUE),
+          nrow = nx, ncol = B)

> bootTestStat <- apply(bootData, 2, median)

> (bootStdErr <- sqrt(var(bootTestStat)))
[1] 0.07937564

> theorStdErr
[1] 0.0677069

> mean(bootTestStat)
[1] 9.796118

> jMedian
[1] 9.804809
```

I conclude ... for a data-generating distribution as bimodal as this, n = 300 is close to -- but not quite in -- Asymptopia.

Good default template for conducting a bootstrap. Can be adapted for other resampling or random data generation tasks.

```
> B <- 1000
```

make it easy to start w/ small B, then scale up

```
> bootData <-
+   matrix(sample(x, size = B * nx, replace = TRUE),
+          nrow = nx, ncol = B)
```

generate the bootstrap data all at once

```
> bootTestStat <- apply(bootData, 2, median)
```

use data aggregation techniques to compute bootstrap statistics

```
> (bootStdErr <- sqrt(var(bootTestStat)))
[1] 0.08163377

> mean(bootTestStat)
[1] 9.796118

> jMedian
[1] 9.804809

> abs(mean(bootTestStat) - jMedian)/bootStdErr
[1] 0.1094916
```

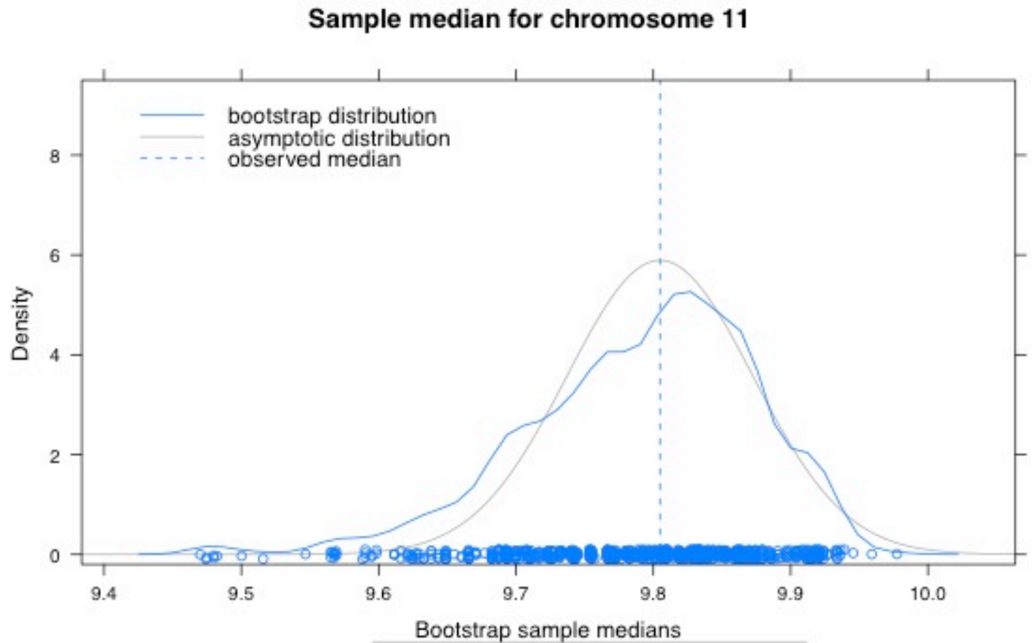
R packages for bootstrapping

- boot:
  - a companion to the book “Bootstrap Methods and Their Applications” by AC Davison and DV Hinkley – seems to be distributed with R
- bootstrap:
  - Companion to the book “An Introduction to the Bootstrap” by Efron and Tibshirani 1993— seems not to be actively maintained.

Using the boot package

boot output

```
> bootRes <- boot(x, function(z, i) median(z[i]), R = 1000)

> bootRes

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data = x, statistic = function(z, i) median(z[i]), R = 1000)

Bootstrap Statistics :
    original       bias    std. error
t1* 9.804809 -0.01221307  0.08194345
```



an interval estimate for the median

```
> boot.ci(bootRes, conf = c(0.90, 0.95), type = "all")
...
Intervals :
Level      Normal              Basic
90%   ( 9.682,  9.952 )   ( 9.692,  9.971 )
95%   ( 9.656,  9.978 )   ( 9.683, 10.019 )

Level     Percentile            BCa
90%   ( 9.638,  9.918 )   ( 9.621,  9.913 )
95%   ( 9.590,  9.927 )   ( 9.579,  9.921 )
```

Bootstrap methods can be used to build CIs. Here showing output from ‘boot’ package, boot.ci() function.

How many bootstrap samples should we generate? (i.e., how large should B be?)

- Efron & Tibshirani recommend B~=200 for the purpose of estimating standard error.

- You will need much more (~1000-10000) for confidence intervals.

- I recommend B=1000 for std error estimation or testing, but why not use much larger B, like B=10,000.

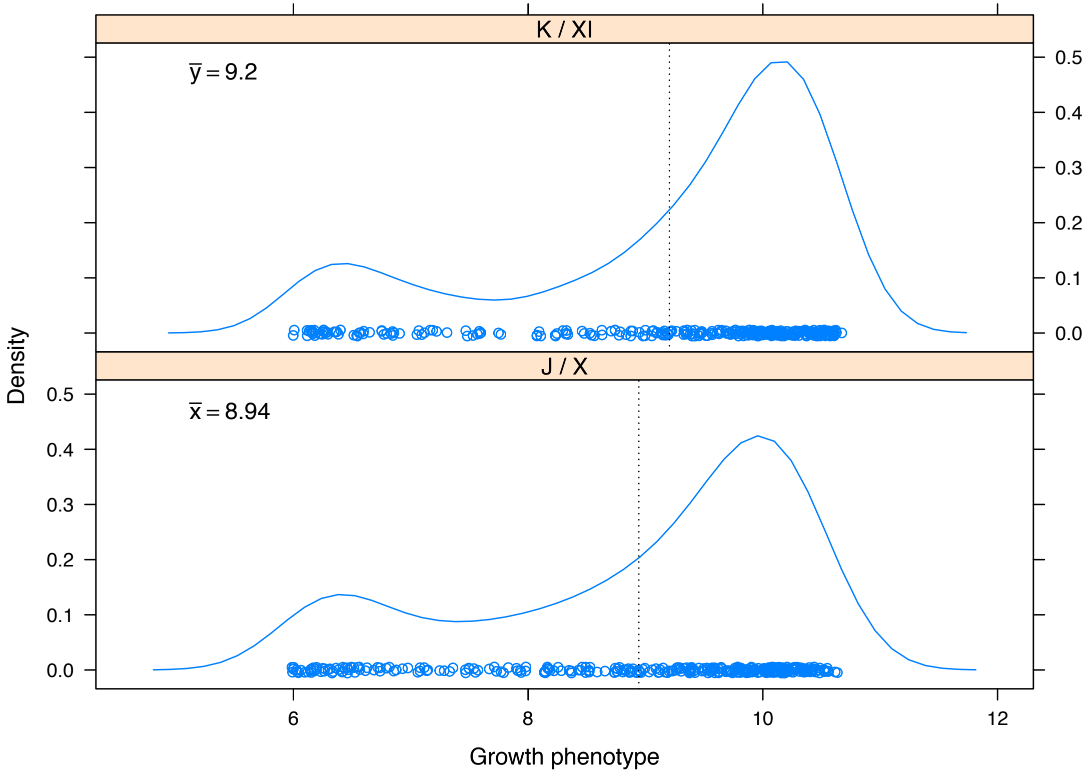
x = data observed from one chromosome, e.g. 10

y = data observed from another chromosome, e.g. 11

Regard x as a realization of X ~ F.

Regard y as a realization of Y ~ G.

F = G?



Specify a null hypothesis H_0: F = G (= H)

x = data observed from one chromosome
y = data observed from another chromosome

Regard x as a realization of $X \sim F$.
Regard y as a realization of $Y \sim G$.

$F = G$?

(biological questions: are the genes on different chromosomes equally important to fitness? is there a relationship between gene location and gene function or essentiality?)

Basics of a hypothesis test

- Specify a null hypothesis, *H_0*
- Choose a test statistic
- Determine the distribution for the test statistic under *H_0*
- Convert the observed test statistic into a p-value

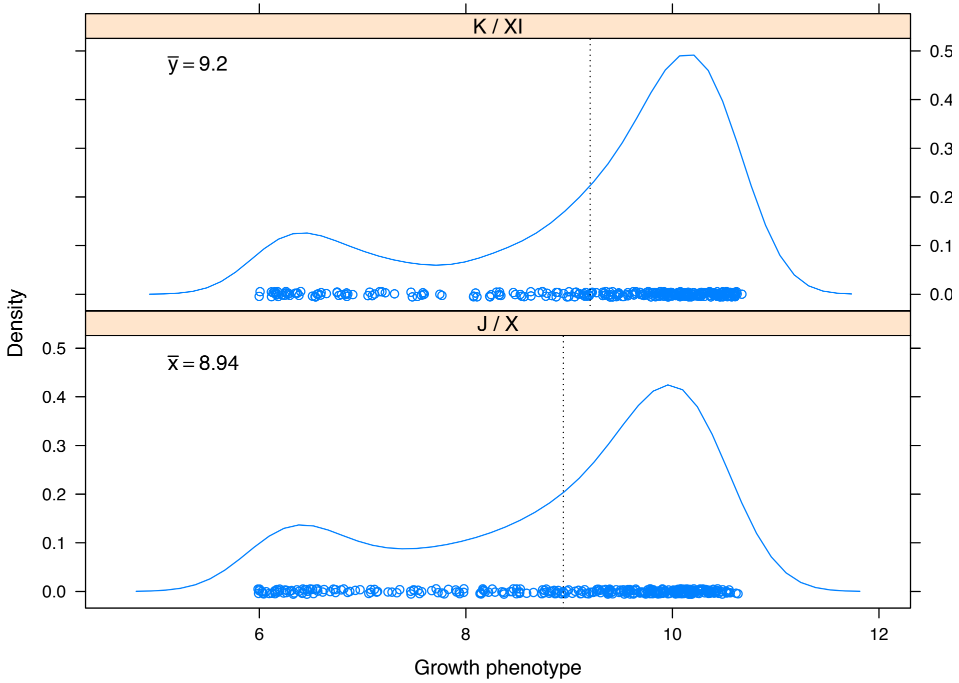
“The p-value is the probability under *H_0* of observing a value of the test statistic the same or more extreme than what was actually observed.”

All of Statistics by Larry Wasserman. Springer, 2004. GoogleBooks search. via myilibrary

All of Nonparametric Statistics by Larry Wasserman. Springer, 2006.
via SpringerLink | via myilibrary | GoogleBooks search.

Classical tests that address our question

- t test
- Wilcoxon test, aka Mann-Whitney here
- Kolmogorov-Smirnov test, 2 sample version
- Chi-square test of homogeneity
- I'm sure there are others ....



Null hypothesis: F = G (= H)

Possible test statistic: |avg (x) - avg (y)|

Observed value of test statistic = t

$$t = \left| \bar{x} - \bar{y} \right|$$

How much evidence does t present against the null hypothesis?

What is the distribution of the test statistic under the null?

Under null, X and Y have same distribution. Let's call it H.

If we knew H, we could draw n_x observations from it -- call this x* -- and another n_y observations from it -- call this y*.

Compute t* = |avg x* - avg y*|.

$$t^* = \left| \overline{x}^* - \overline{y}^* \right|$$

Compute t* = |avg x* - avg y*|.

$$t^{*} = \left|\overline{x}^{*} - \overline{y}^{*}\right|$$

Generate B such observations t* (B large).

What proportion of the t* are as or more extreme as t? That's basically your bootstrap p-value.

Done! Sort of. We don't actually know H, though.

Here we can estimate H with an empirical distribution function.

Amalgamate x and y into one sample. Under the null, they are iid H. Give mass 1/(n_x + n_y) to each observation. That's the empirical distribution function. That's a decent estimate of H.

How to generate data from this estimate of H? Resample with replacement.

Choose a test statistic

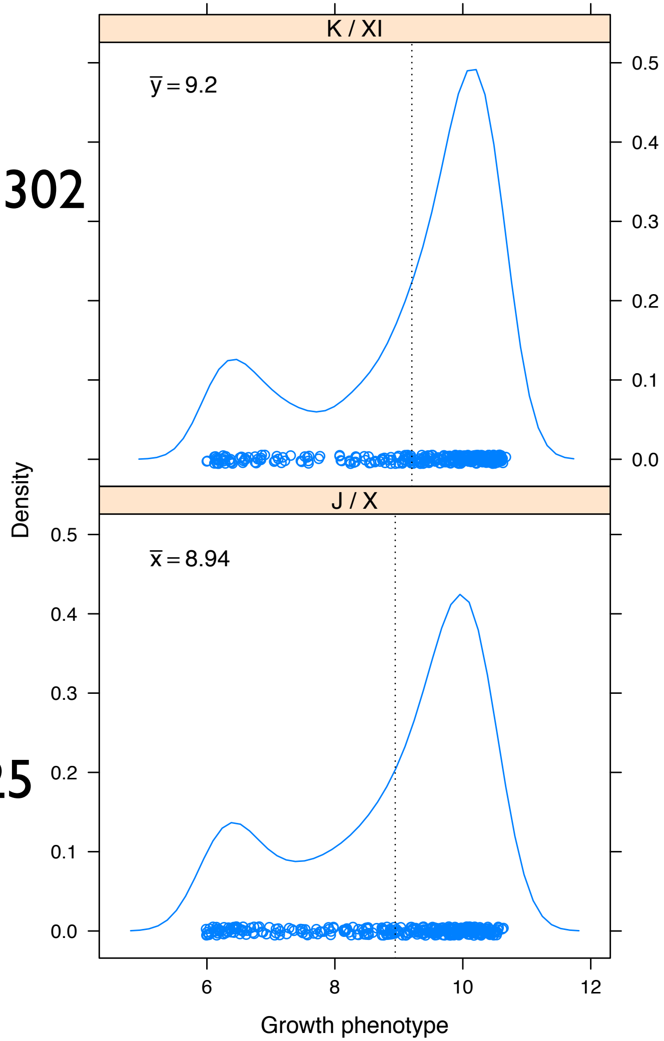
Let's try this: $t = |\bar{x} - \bar{y}|$

```
> (chromoMeans <- with(kDat,
+                     tapply(pheno, chromo, mean)))
      10        11
8.943558 9.203379

> (obsTestStat <- abs(chromoMeans[1] - chromoMeans[2]))
       10
0.2598215
```

$$\bar{x} = 8.94$$

$$\bar{y} = 9.2$$

$$t = |\bar{x} - \bar{y}| = 0.26$$



$n_y = 302$

$n_x = 325$

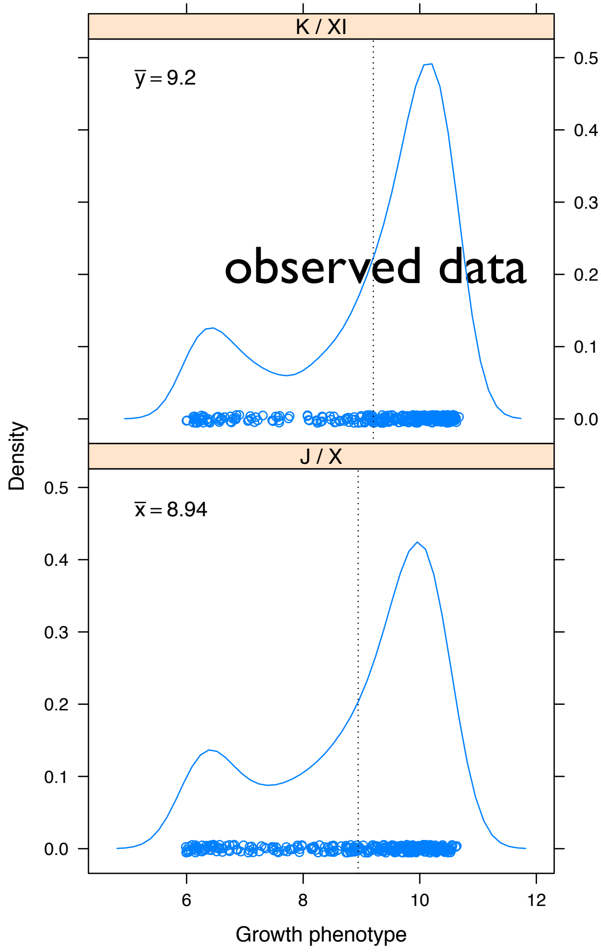
$$t = \left| \overline{x} - \overline{y} \right| = 0.26$$

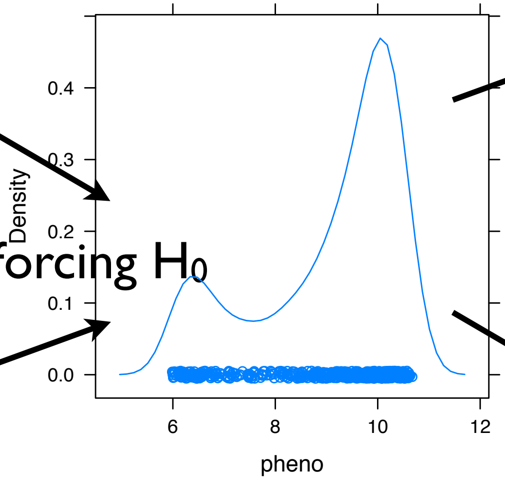
Is this “big” or “extreme” and, therefore, suggests we should reject H_0?

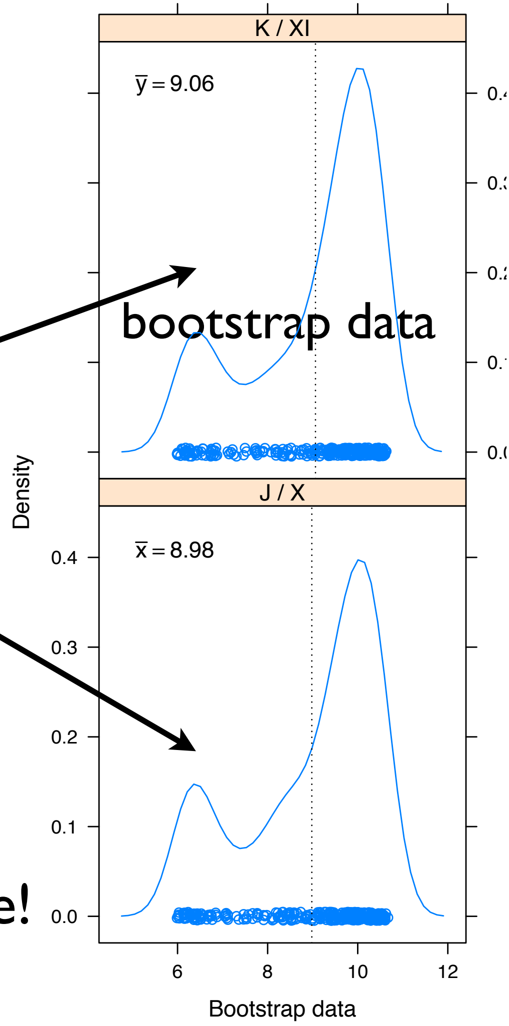
Ideally, we would generate lots of datasets from the (unknown) distribution H and get an empirical null distribution for this test statistic. But we don’t know H ......

```
> (obsTestStat <- abs(chromoMeans[1] - chromoMeans[2]))
       10
0.2598215

<snip, snip>

> (bootTestStat <- abs(diff(bootMeans)))
        11
0.08155161
```

1 bootstrap sample (“baby steps”)

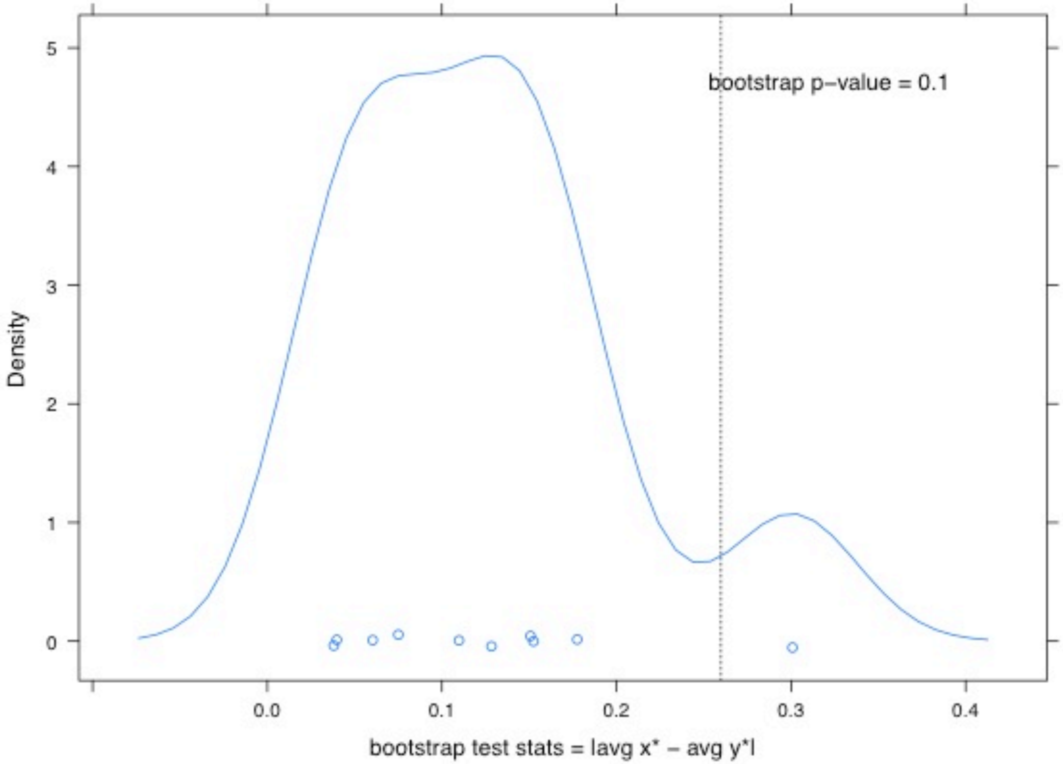

observed data


enforcing H_0


bootstrap data

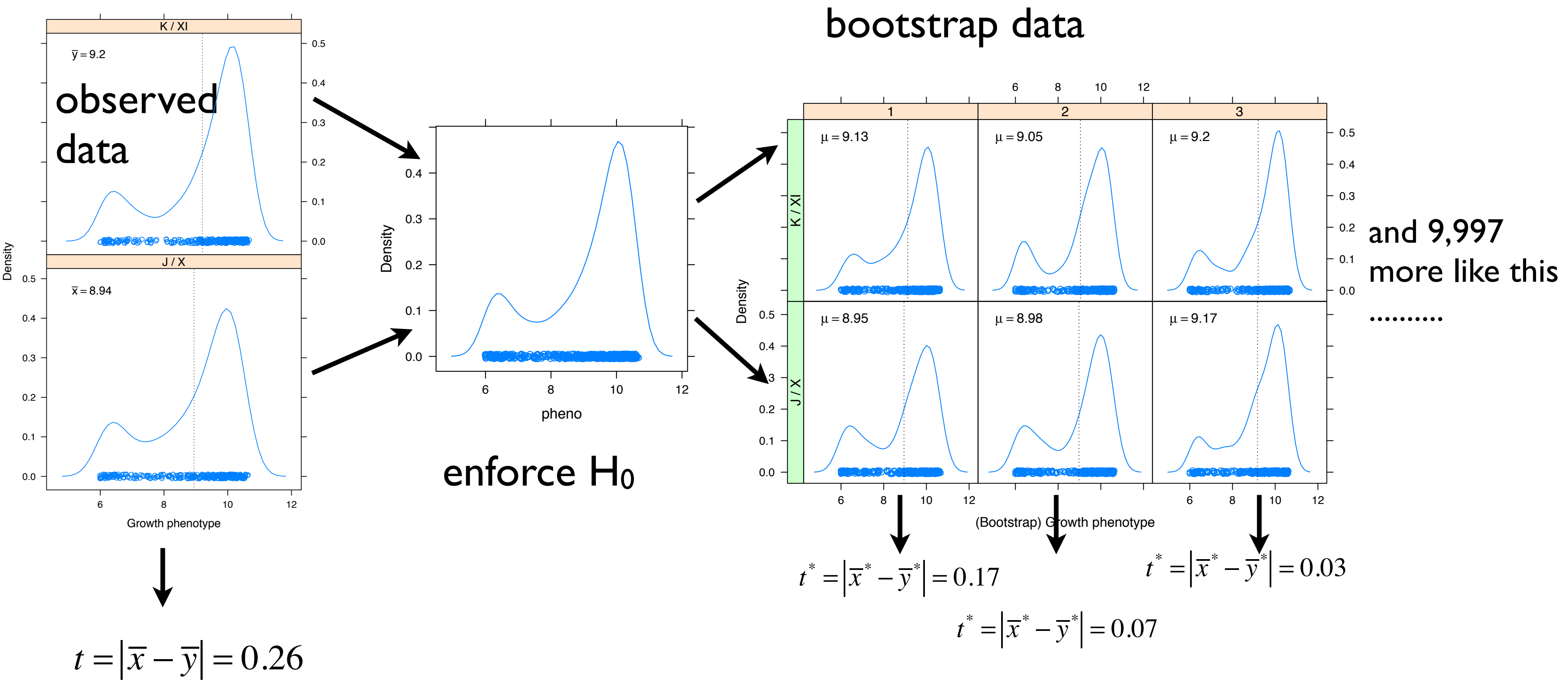
So far, the observed test statistic looks pretty extreme! Let’s scale up a bit ....

```
> (obsTestStat <- abs(chromoMeans[1] - chromoMeans[2]))
       10
0.2598215

...
> bootTestStat
 [1] 0.23677776 0.21074474 0.16380568 0.13258165 0.01663695 0.07176389
 [7] 0.09824504 0.20745668 0.11928580 0.27759690
> mean(bootTestStat >= obsTestStat)
[1] 0.1
```



What proportion of the t* are as or more extreme as t? That's basically your bootstrap p-value.

bootstrap data



observed data

enforce H_0

and 9,997 more like this

....

$$t = |\bar{x} - \bar{y}| = 0.26$$

$$t^* = |\bar{x}^* - \bar{y}^*| = 0.17$$

$$t^* = |\bar{x}^* - \bar{y}^*| = 0.07$$

$$t^* = |\bar{x}^* - \bar{y}^*| = 0.03$$

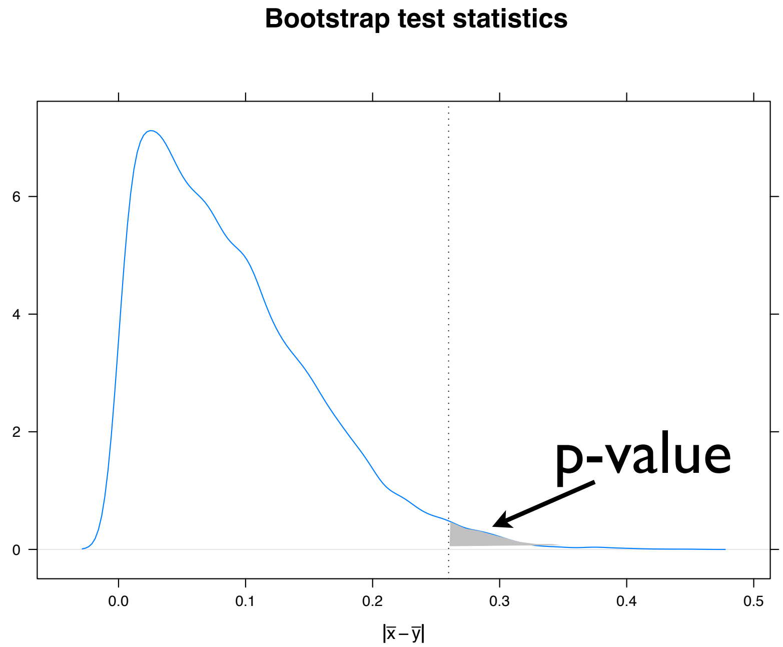
```
(n <- nrow(kDat))                       # 627 obs
B <- 10000
bootDat <-
  matrix(sample(kDat$pheno, size = B * n, replace = TRUE),
         nrow = n, ncol = B)
str(bootDat)
## num [1:627, 1:10000] 10.08 10.07 10.03 9.26 8.28 ...
bootTestStats <-
  apply(bootDat, 2, computeAbsDifferenceOfMeans, jFact = kDat$chromo)
```

No explicit loops!

B = 10,000 bootstrap samples

```
> bootTestStats <-
+         apply(bootDat, 2, computeAbsDifferenceOfMeans, jFact = kDat$chromo)

> densityplot( ~ bootTestStats,
+             xlab = expression(group("|", bar(x) - bar(y),"|")),
+             main = "Bootstrap test statistics",
+             plot.points = FALSE, n = 200, ref = TRUE,
+             panel = function(x, ...) {
+               panel.densityplot(x, ...)
+               panel.abline(v = obsTestStat, lty = 'dotted')
+             })

> ## bootstrap p-value
> mean(bootTestStats >= obsTestStat)
[1] 0.0172

> t.test(pheno ~ chromo, kDat)$p.value
[1] 0.01940612
```

Bootstrap p-value is very close to Welch's t-test p-value. That's comforting!



Permutation test in hypothesis testing

- Most commonly used resampling method for hypothesis testing.
- Sample without replacement your response (and/or group memberships) – e.g., permute the labels.
- In theory enables us to compute exact p-values.

Stat. Appl. Genet. Molec. Biol. Volume 9, Issue 1, Article 39.

Simple example: differential gene expression analysis

- Suppose we want to find genes that are differentially expressed between different conditions.
- We compute the test statistic (e.g., t-statistics) for each of the g genes.
- We **have** the distribution of the test statistic under the null.
- P-value quantifies the probability of observing the observed test statistic under the null model.

Standard t-test

- Assume $X_1, X_2, \ldots, X_m$ are from ~ $N(\mu_1, \sigma^2)$
- Assume $Y_1, Y_2, \ldots, Y_n$ are from ~ $N(\mu_2, \sigma^2)$

- Compute the pooled variance estimate:

$$s^2 = \frac{1}{m+n-2}(\sum_{i=1}^{m}(X_i - \bar{X})^2 + \sum_{i=1}^{n}(Y_i - \bar{Y})^2).$$

- The t-statistic is given by

$$T(X,Y) = \frac{\bar{X} - \bar{Y}}{s\sqrt{\frac{1}{m} + \frac{1}{n}}}.$$

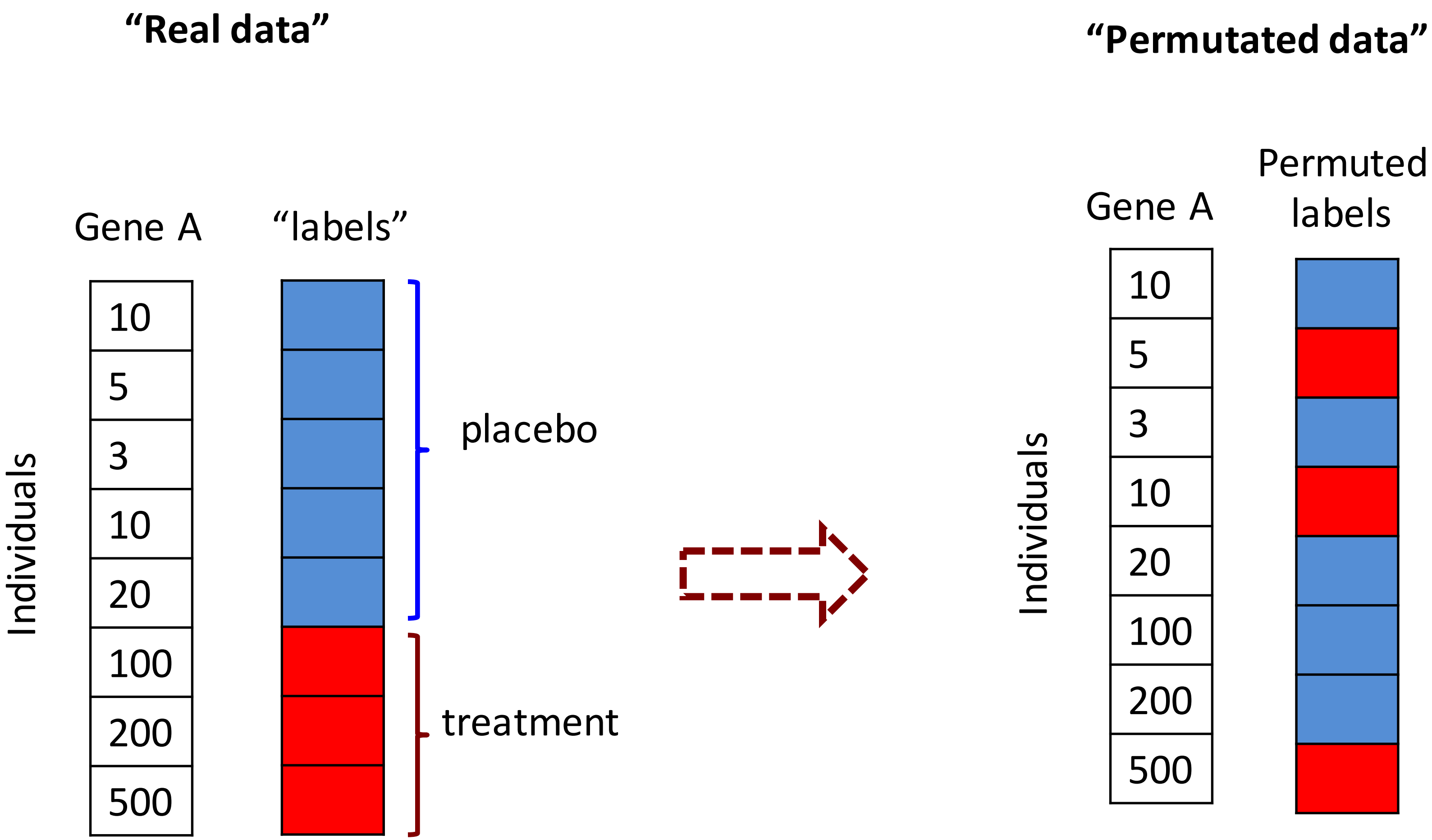
Permutation test

- Want to test whether observation in two groups follows the same distribution, without making assumptions about the distributions (e.g., normality)
- Generate a null distribution for the test-statistic:
  - Randomly divide individuals to ‘treatment’ groups
- For i = 1 …. p, do
  - Permute the group labels, giving new assignment of ‘group; to each individual
  - Computer the test statistic from the permutated data

“Real data”

Individuals

Gene A	“labels”	
10	0	placebo
5	0	
3	0	
10	0	
20	0	
100	1	treatment
200	1	
500	1	



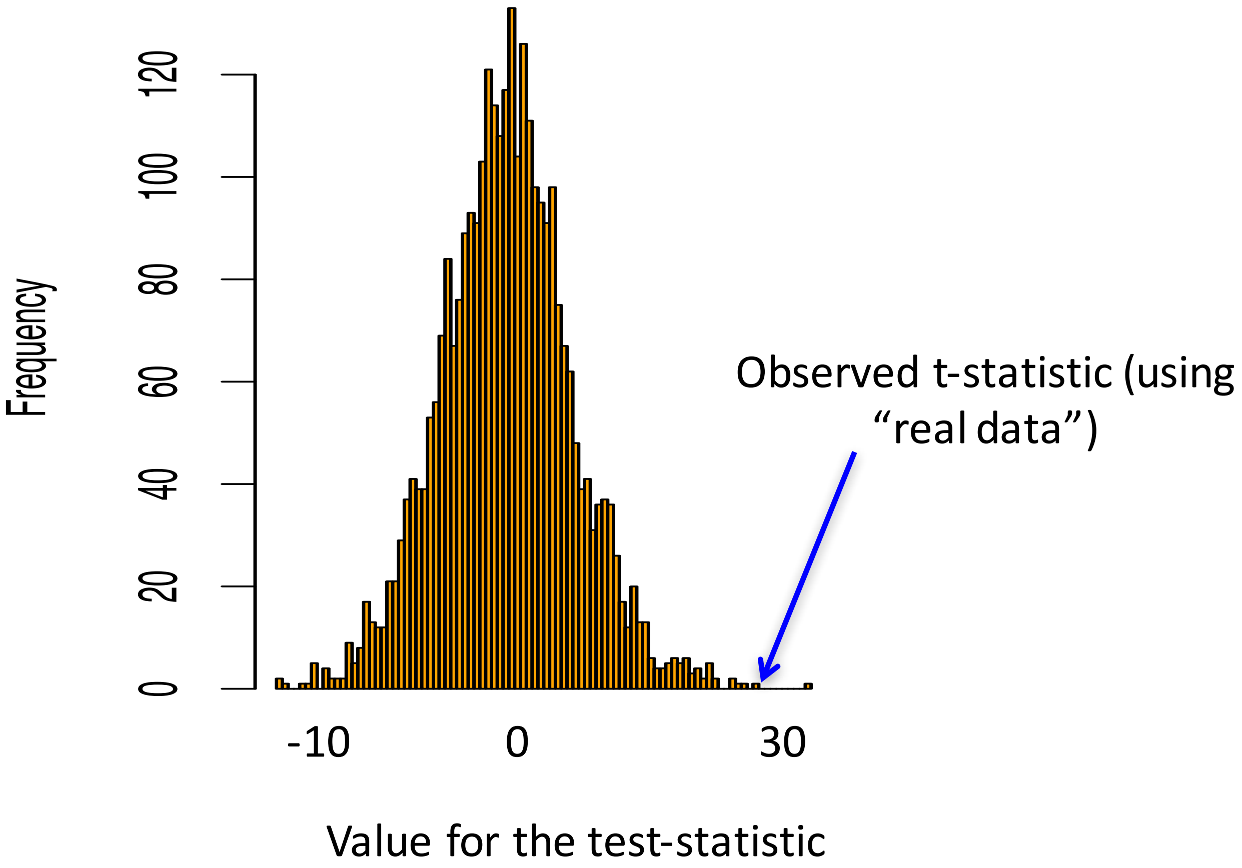
“Real data”

Gene A	“labels”	
10	🟦	placebo
5	🟦	
3	🟦	
10	🟦	
20	🟦	
100	🟥	treatment
200	🟥	
500	🟥	

“Permutated data”

Gene A	Permuted labels
10	🟦
5	🟥
3	🟦
10	🟥
20	🟦
100	🟦
200	🟦
500	🟥

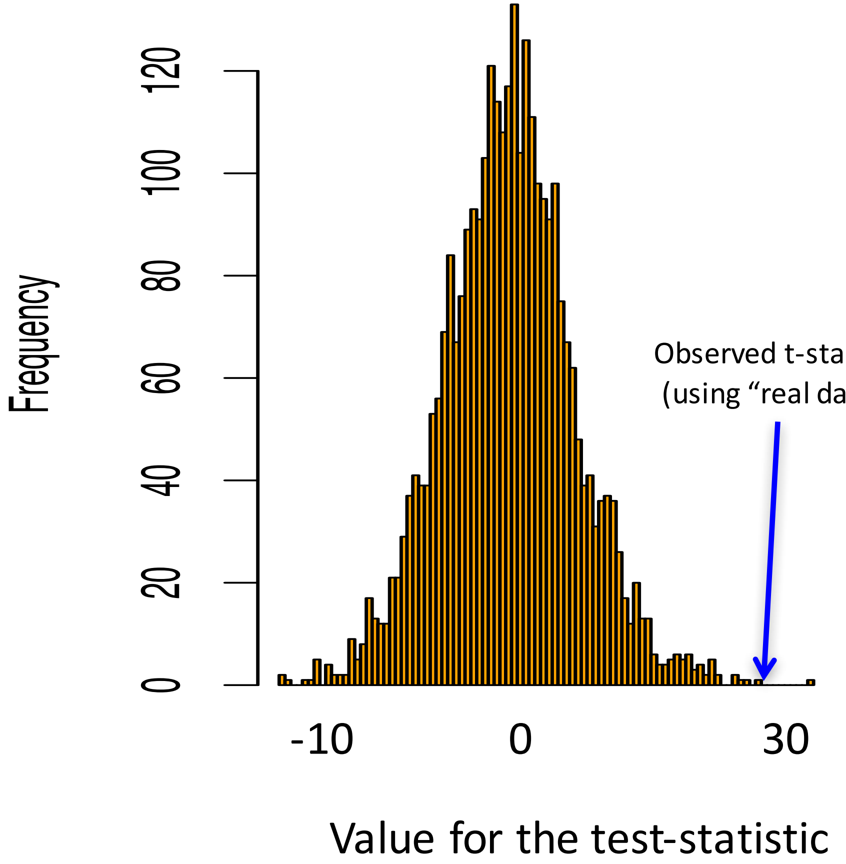
Histogram of test-statistic under null (permutated data)



Observed t-statistic (using “real data”)

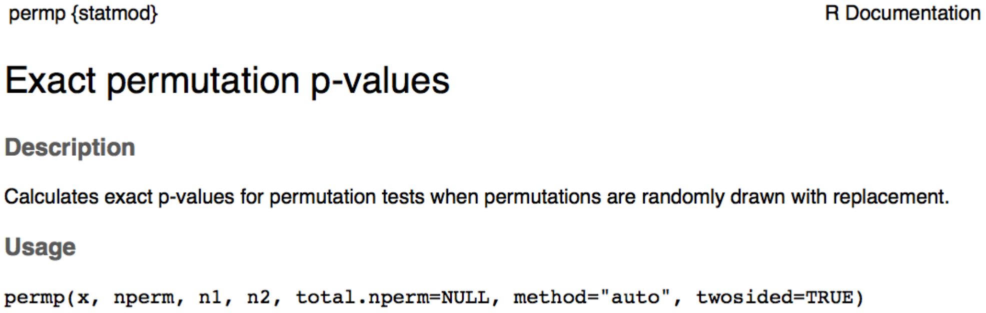
Histogram of test-statistic under null (permutated data)

The null distribution for $\bar{X} - \bar{Y}$



Observed t-statistic
(using “real data”)

P-value: $$\frac{\#(T_p > T_r)}{\# \, permutations}$$



permp {statmod} R Documentation

Exact permutation p-values

Description

Calculates exact p-values for permutation tests when permutations are randomly drawn with replacement.

Usage

```
permp(x, nperm, n1, n2, total.nperm=NULL, method="auto", twosided=TRUE)
```

Resampling methods

- Ways of performing statistical inference that are "internal to the data" under analysis: e.g., you get the necessary knowledge about sampling variability (of parameters/estimates) from the observed data itself.

- Resampling methods:
  - Bootstrap
  - Permutation testing
  - Cross validation