Quantitative Applications for Data Analysis: Distributions & Hypothesis Testing

Erik Spence

SciNet HPC Consortium

6 February 2024

Today's slides

Today's slides can be found here. Go to the "Quantitative Applications for Data Analysis" page, under Lectures, "Distributions".

`https://scinet.courses/1346`

Today's class

Today we are going to begin our adventure in the world of statistics.

- Distributions.
- R functions for calculating distributions.
- Confidence intervals.
- Hypothesis testing.
- Type I and type II errors.
- The *apply family of functions.

As with all classes, please stop me if you have a question.

Statistics

To begin at the beginning: what is statistics?

- Statistics is a collection of techniques for empirically describing populations, collections of populations, and the relationships between them.
- Usually we do not have the complete population at our disposal, we only have a sample of the population.
- We use this sample to draw conclusions about the true population from which the sample was drawn, assuming that the sample is representative of the whole population.
- We also use the sample to perform tests to determine the relationship between different populations.

Right. No problem.

Data come from distributions

We usually use a probability distribution to model our data. What is a probability distribution?

- A probability distribution indicates the probability of a given event (or measurement, or observation) happening.
- There are two types of probability distributions:
  - discrete: data come in individual steps, there are no data points between those steps (flips of a coin, rolls of dice).
  - continuous: data are real numbers, with decimal places.
- All probability distributions have the following properties:
  - The sum of all probabilities equals one.
    - ★ Discrete: $\sum_i P(x_i) = 1$
    - ★ Continuous: $\int \rho(x) dx = 1$
  - One minus the probability of something is the probability of not something. $P'(x) = 1 - P(x)$

Discrete distributions

Discrete distributions apply when the data are not continuous, the data come in discrete steps.

Binomial distribution:

$$P(x=k) = \binom{n}{k} p^k \left(1-p\right)^{n-k}$$

where p is the probability of success, k is the number of successes and n is the number of attempts.

An example of this would be picking n marbles out of a bag of red and black marbles, and picking k red marbles.

Coin toss:

$$P(x) = \begin{cases} 0.5, & x = \text{heads}, \\ 0.5, & x = \text{tails} \end{cases}$$

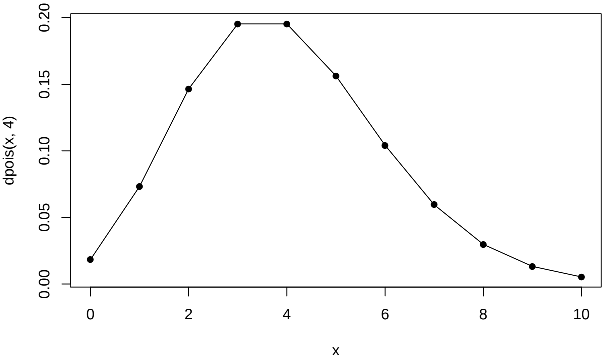
Roll of a die:

$$P(x) = \begin{cases} 1/6, & x = 1, \\ 1/6, & x = 2, \\ 1/6, & x = 3, \\ 1/6, & x = 4, \\ 1/6, & x = 5, \\ 1/6, & x = 6 \end{cases}$$

Poisson distribution

The Poisson distribution is used for discrete events that happen infrequently. The probability of the event happening must increase with time.

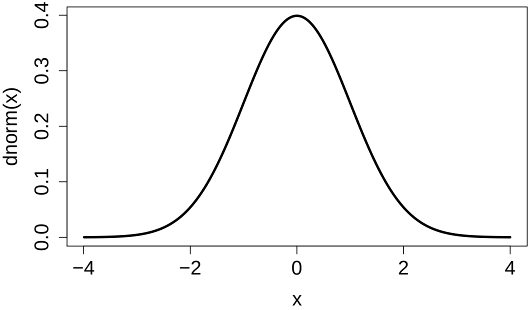
Please do not use continuous distributions for discrete variables!

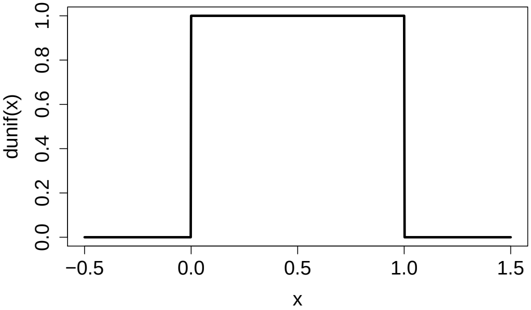


Poisson distribution $(\lambda = 4)$:

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

Continuous distributions





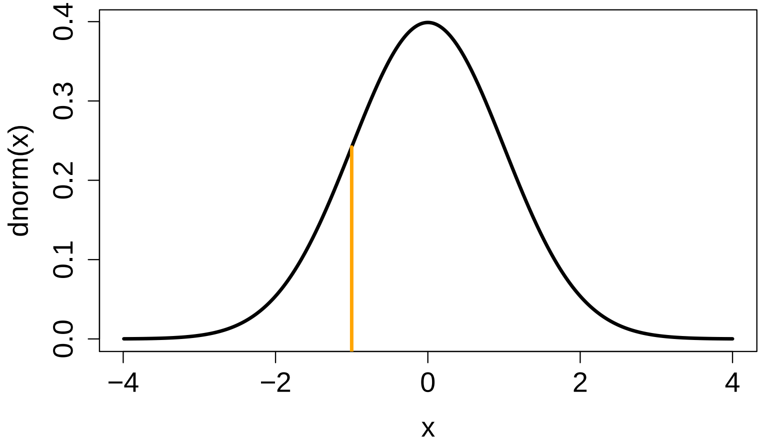
$$\rho(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Gaussian

$$\rho(x) = \begin{cases} 1, & 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

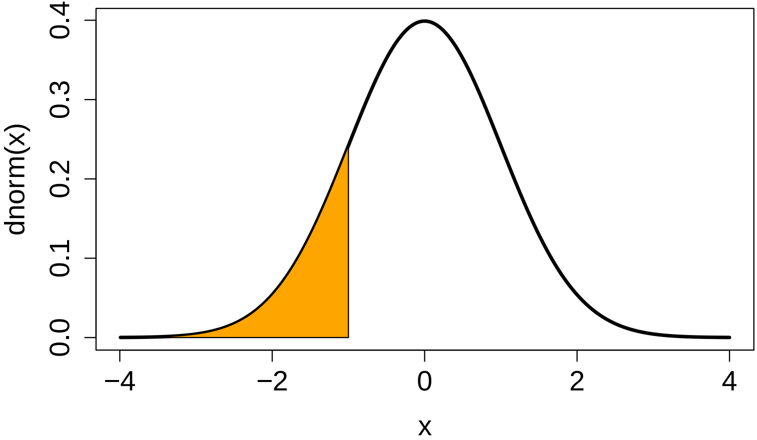
Uniform

Some distribution terminology



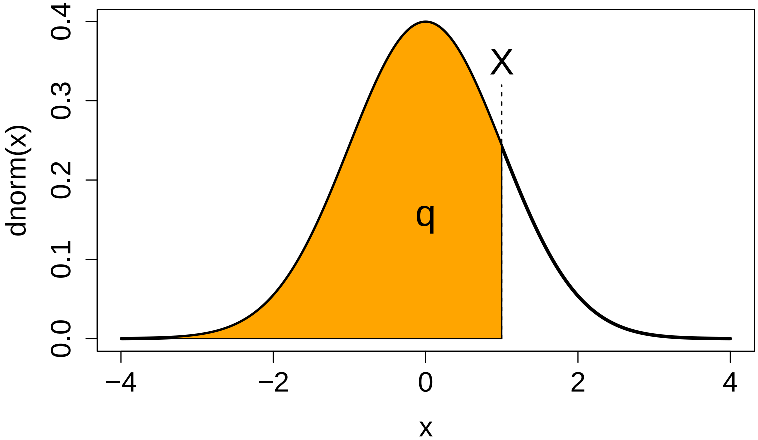
Probability density function (PDF), $\rho(x)$: the function which determines probability of getting a particular value, X, from a continuous distribution, $P(x = X) = \rho(x)dx$.

Some distribution terminology, continued



Cummulative distribution function (CDF): the probability of getting a particular value of x below a certain value, $P(x < X) = \int_{-\infty}^{X} \rho(x) dx$.

Some distribution terminology, continued more



Quantile function (QFn): given a probability q, the particular value of X such that $P(x < X) \leq q$.

Some more distribution terminology

Some terms you're all likely aware of.

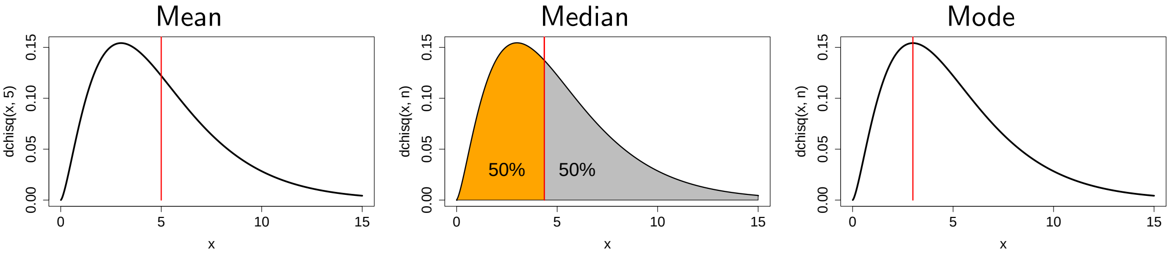
- the Expectation value: $\langle f \rangle = E(f(x)) = \int_{-\infty}^{\infty} f(x)\rho(x)dx$.
- the mean: $\langle x \rangle = \int_{-\infty}^{\infty} x\rho(x)dx$.
- variance: $\sigma^2(x) = \left\langle (x - \langle x \rangle)^2 \right\rangle$
- Standard deviation: square root of the variance.

R comes with many built-in functions to calculate the usual quantities.

- mean(), sd(), var()
- min(), max()
- range()
- summary()

As usual, type "help(mean)" to learn how these functions can be used. Do NOT use these function names as variables!

Characteristic statistics



A warning: many distributions are not centred. If the distribution of your data is not centred the concept of a 'mean' may not be meaningful.

The chi-squared distribution, shown here, is not centered or symmetric around its peak.

Built-in datasets, an aside

R contains built-in datasets that can be used for practicing.

```
> data()
Data sets in package ‘datasets’:

 AirPassengers             Monthly Airline Passenger Numbers 1949-1960
 BJsales                   Sales Data with Leading Indicator
 BJsales.lead (BJsales)    Sales Data with Leading Indicator
 BOD                       Biochemical Oxygen Demand
  ⋮
--------------------------------------------------------------------------------
>
--------------------------------------------------------------------------------
> str(faithful)
data.frame':   272 obs.  of 2 variables:
$ eruptions: num 3.6 1.8 3.33 2.28 4.53 ...
$ waiting  : num 79 54 74 62 85 55 88 85 51 85 ...
--------------------------------------------------------------------------------
>
```

Type 'q' to get out of the 'data' menu. DO NOT use 'data' as a variable.

R distribution functions

R has a tonne of distributions built into it. The syntax for using the distributions is fairly consistent. To access a particular distribution, you use the following suffixes:

- Uniform: unif
- Normal: norm
- Binomial: binom
- Poisson: pois
- and many many others

To access particular functions associated with those distributions, you use the prefixes:

- Probability Distribution Function (PDF): d
- Cummulative Distribution Function (CDF): p
- Quantilve Function (QFn): q
- Random sampling from the distribution: r

R distribution functions, continued

Distribution	suffix	PDF (d)	CDF (p)	QFn (q)	Sample (r)
Normal	norm	dnorm	pnorm	qnorm	rnorm
Uniform	unif	dunif	punif	qunif	runif
Exponential	exp	dexp	pexp	qexp	rexp
Poisson	pois	dpois	ppois	qpois	rpois
Binomial	binom	dbinom	pbinom	qbinom	rbinom

Note that most distributions take optional arguments which control their behaviour (use the 'help' function to get details on how to use these functions).

R distribution functions, examples

```
>
> # Normal distribution probabilities with
> # default values (mean = 0, sd = 1)
> dnorm(c(-2, 0, 2))
[1] 0.05399097 0.39894228 0.05399097
>
> # Normal distribution probabilities with
> # mean = 1, sd = 2
> dnorm(c(-2, 0, 2), mean = 1, sd = 2)
[1] 0.0647588 0.1760327 0.1760327
>
> # Value of normal distribution which
> # has a probability of 0.025
> qnorm(0.025)
[1] -1.959964
>
```

```
>
> # 4 random samples from the normal distribution
> # with default values (mean = 0, sd = 1)
> rnorm(4)
[1] -0.01890732 -1.51366406 1.00561637 -0.27690594
>
> # 4 random samples from the normal distribution
> # with mean = 1, sd = 2
> rnorm(4, mean = 1, sd = 2)
[1] 1.70841356 2.96691253 0.07857346 -0.87288538
>
> # 4 random samples from the Poisson distribution
> # with lambda = 20
> rpois(4, lambda = 20)
[1] 26 19 25 15
>
```

Applied examples

Suppose that a given collection of insects have weights that are normally distributed with a

- mean of 17.46 grams and
- variance of 75.67 grams^2.

What is the probability that a randomly chosen insect within the collection weighs more than 19 grams?

We follow these steps:

- We use the cummulative distribution function (CDF) to get the probability of being less than 19 grams.
- We then subtract this from 1 to get the probability of being greater than 19 grams.

```
>
> p <- pnorm(19, mean = 17.46, sd = sqrt(75.67))
>
> 1 - p
[1] 0.4297405
>
```

Answer: 43%

Applied examples, continued more

Suppose some widgets produced at the Acme Factory have a probability of 0.005 of being defective. The widgets are shipped in boxes of 25.

- What is the probability that a box contains exactly 1 defective widget?
- What is the probability that a box has *at most* 1 defective widget?

Use a binomial distribution, since each widget is either defective or not. The binomial distribution takes two optional arguments:

- size (the number of samples),
- prob (the probabilty of something occuring).

```
>
> dbinom(1, size = 25, prob = 0.005)
[1] 0.1108317
>
> pbinom(1, size = 25, prob = 0.005)
[1] 0.9930519
>
```

Answer 1: 11% Answer 2: 99%

Sample means, continuous variables

It's very important to distinguish between sample statistics and population statistics. The population is what we want to describe, but the sample from the population is what we have.

We often want to get statistics about the mean of our sampled data, $\bar{x}$. For continuous variables:

- What is mean of the sampled means? Meaning, if we sampled from the population many times, what would the mean of all the sampled subpopulation means be? Well, it turns out to be just μ, the population mean.
- What is the variance of the sampled means? It's σ^2/n, where n is the number of samples.

Which means, of course, that the standard deviation of the sampled mean is $\sigma/\sqrt{n}$. This means that the more samples we have the better our estimate of the population mean will be.

Sample means, continuous variables, continued

The previous slide assumed that we know σ^2, the population variance. How do we get that?

- We should use the sample variance, $s^2 = \frac{\sum(x_i - \bar{x})^2}{n-1}$.
- Why $n - 1$, instead of n?
- That's what gives us σ^2, meaning s^2 is an "Unbiased Estimator".
- It turns out that s is not an Unbiased Estimator, but it's still an acceptable estimation of σ.
- Note that the "var" command uses $n - 1$ in its denominator.

```
> sample.data <- function(n, sdev, div) {
+     my.data <- rnorm(n, sd = sdev)
+     return(sum((my.data - mean(my.data))**2) / div)
+ }
>
> m <- 10000
> result <- rep(0, m)
>
> for (i in 1:m) result[i] <- sample.data(10, 3, 9)
> mean(result)
[1] 16.004043
>
> for (i in 1:m) result[i] <- sample.data(10, 3, 10)
> mean(result)
[1] 14.34681
>
```

Calculating confidence intervals

Suppose that you've calculated the mean, $\bar{x}$, of some samples. What is the uncertainty on that μ?

To answer this question, we estimate the Standard Error

$$\mathbf{SE}\left(\bar{x}\right)^2 = \frac{s^2}{n}$$

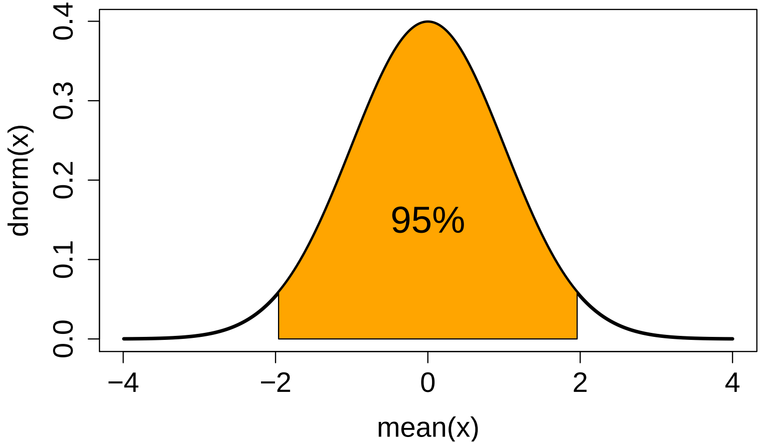
The 95% confidence interval, in which there is a 95% chance the true mean of the population lies, is given by

$$\mu \pm 1.96\ \mathbf{SE}(\bar{\mathrm{x}})$$

This is because 1.96 standard deviations is approximately what contains 95% of the normal distribution.

```
>
> x <- trees$Girth
>
> my.mean <- mean(x)
>
> my.mean
[1] 13.24839
>
> se2 <- var(x) / length(x)
>
> my.mean - 1.96 * sqrt(se2)
[1] 12.14368
>
> my.mean + 1.96 * sqrt(se2)
[1] 14.35309
>
```

Calculating confidence intervals



We select a confidence interval that includes 95% of the area under the Gaussian.

Hypothesis testing

How do I perform statistical tests on my data?

- Statistical tests are always testing *against* something.
- The thing being tested against is called the *Null Hypothesis*, $\boldsymbol{H_0}$.
- All hypothesis testing is done under the assumption that the Null Hypothesis is true.
- The non-Null Hypothesis is called the Alternate Hypothesis, $\boldsymbol{H_1}$.
- Every hypothesis test is attempting to answer the question: Should I reject the null hypothesis?

You may have heard of the null hypothesis before. What are the characteristics of the null hypothesis?

- It represents NO change from the accepted state of things.
- Whatever is 'normal', or 'default', is the null hypothesis.

Hypothesis testing, example

Step 1: make a claim: "He's dead, Jim", said Dr. McCoy to Captain Kirk. Under usual circumstances this claim will involve some test statistic (mean, variance, *etc.*).

Step 2: determine if the claim being made is the null or alternative hypothesis.

Does this statement represent a change from the normal situation?

- If yes, then it is the alternate hypothesis, $\boldsymbol{H_1}$.
- If not, then it is the null hypothesis (the normal situation is that he is not dead), $\boldsymbol{H_0}$.

Step 3: make a decision (perform a test) to determine whether the null hypothesis should be rejected or not rejected.

- Reject $\boldsymbol{H_0}$: "sufficient evidence to say the patient is dead".
- Fail to reject $\boldsymbol{H_0}$: "insufficient evidence to say patient is dead".

Note that we never accept the null hypothesis, we merely fail to reject it.

Example stolen from James Jones.

Hypothesis testing, example, continued

Based on the two possible states (dead/alive) and the two possible decisions (reject $\boldsymbol{H_0}$/fail to reject $\boldsymbol{H_0}$), there are 4 possible outcomes.

	True state of nature	
Decision	$\boldsymbol{H_0}$ True (patient is not dead)	$\boldsymbol{H_0}$ False (patient is dead)
Reject $\boldsymbol{H_0}$	Patient is not dead, Sufficient evidence of death	Patient is dead, Sufficient evidence of death
Fail to reject $\boldsymbol{H_0}$	Patient is not dead, Insufficient evidence of death	Patient is dead, Insufficient evidence of death

Or, in other words...

	True state of nature	
Decision	$\boldsymbol{H_0}$ True	$\boldsymbol{H_0}$ False
Reject $\boldsymbol{H_0}$	Dispose of a live person	Dispose of a dead person
Fail to reject $\boldsymbol{H_0}$	Try to revive a live person	Try to revive a dead person

Hypothesis testing, example, continued more

	True state of nature	
Decision	H_0 True	H_0 False
Reject H_0	Dispose of a live person	Dispose of a dead person
Fail to reject H_0	Try to revive a live person	Try to revive a dead person

These cases are so common (and well-studied) that they have been given names.

	True state of nature	
Decision	H_0 True	H_0 False
Reject H_0	Type I error (alpha)	Correct assessment
Fail to reject H_0	Correct assessment	Type II error (beta)

Type I errors are usually considered more serious.

How hypothesis tests work

Suppose we ask the question: does a certain antibiotic work at killing bacteria, in the lab?

- The null hypothesis is that the antibiotic does not affect the bacteria population.
- All measurements have errors, and there will be variations (randomness) in the data, so even if there is no effect the average bacteria population differences between the two measurements (before and after antibiotic application) will be non-zero.
- We only have a real difference in bacteria populations if the difference we see is *unlikely to occur by chance* if the real difference is zero.
- The probability of a difference as big as we measure occurring when the null hypothesis of no difference is true is called p-value. This is calculated with the test statistic:

$$t = \frac{\overline{x} - 0}{s/\sqrt{n}}$$

where s is the sample standard deviation, and x is the difference between the population measurements, for a given sample, before and after antibiotic application.

How hypothesis tests work, continued

Due to randomness, we expect $\overline{x}$ to have an approximate Gaussian distribution, with $\mu = 0$ if the null hypothesis is true.

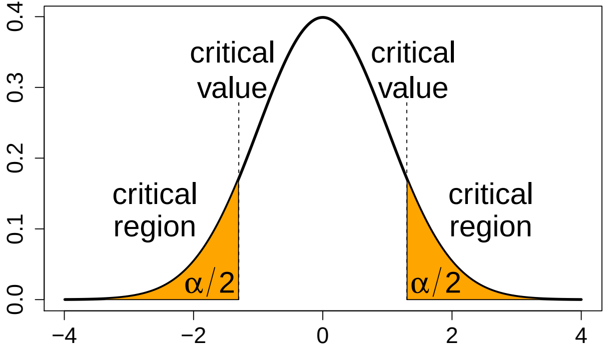
- Using our data, we calculate t.
- This tells us how many standard deviations away from 0 our data is.
- Using this, we can determine the probability that this value of t, or greater, would occur if the null hypothesis is true.
- This is our p-value.
- This is equivalent to determining the probability of committing a Type I error (incorrectly rejecting the null hypothesis).

For those familiar with these sorts of tests, this is an example of a paired two-sample t-test.

Test significance, continued

The significance level (alpha) is used to determine if the test statistic is far enough out in the tails to reject the null hypothesis.

- Commonly used values of significance are 0.05 and 0.01.
- Most tests return a "p-value". This is the probability of commiting a Type I error, given the input data.
- If the p-value is less than the significance, then the null hypothesis can be rejected.



You must decide what significance you are using before you run the test!

How to perform a hypothesis test

Ok, so you want to perform a hypothesis test on your data. What are the steps involved?

- Write the claim, and determine whether it is the null or alternate hypothesis.
- Choose the level of significance (α).
- Perform the test.
- Reject or fail to reject the null hypothesis.
- Write a conclusion.

We've already discussed determining the type of hypothesis from the claim, and the significance level.

What distribution describes my data?

Suppose you've got some data, and you're curious as to the distribution which generates it. Maybe it's Gaussian?

Tests exist to determine whether a set of data is likely from a given distribution. Tests for the normal distribution include:

- Shapiro-Wilk (shapiro.test)
- Anderson-Darling (ad.test)
- Lilliefors (lillie.test)
- Pearson Chi-square (pearson.test)

These tests usually take the null hypothesis to be the case that the data IS normally distributed. Nonetheless, always be sure to read the documentation to confirm what the null hypothesis is. Otherwise, you won't know the p-value is referring to.

What distribution describes my data?, continued

Suppose that we are examining the 'trees' data set.

The null hypothesis is that the data IS normally distributed.

The p-value from two different tests suggest that the null hypothesis cannot be rejected.

Note that many of the tests for normality are found in the 'nortest' library.

```
>
> shapiro.test(trees$Height)

	Shapiro-Wilk normality test

data:  trees$Height
W = 0.96545, p-value = 0.4034
>
> library(nortest)
>
> ad.test(trees$Height)

	Anderson-Darling normality test

data:  trees$Height
A = 0.35926, p-value = 0.4282
>
```

What distribution describes my data?, continued more

If you're not sure what the null hypothesis is, you can always test it by testing the test.

If you go down this route, be sure to run the tests many times, to make sure you don't accidentally stumble upon a case that incorrectly rejects the null hypothesis.

In this case the null hypothesis that the data is normally distributed, when the data is drawn from the uniform distribution, can be confidently rejected.

```
>
> shapiro.test(rnorm(1000))

     Shapiro-Wilk normality test

data:  rnorm(1000)
W = 0.99882, p-value = 0.767
>
> ad.test(runif(1000))

     Anderson-Darling normality test

data:  runif(1000)
A = 12.326, p-value < 2.2e-16
>
```

The *apply family of functions

The *apply family of functions make it very easy and fast to repeatedly apply a function to a set of individual elements.

Many parallel routines are parallel versions of these higher-level functions.

- lapply: apply a function to each element of a list or vector.
- sapply: like lapply, but return a vector instead of a list.
- apply: apply a function to rows, columns or elements of an array.
- tapply: apply a function to subsets of a list or vector.
- mapply: apply a function to the "transpose" of a list.

Using these functions, rather than regular for loops, can significantly speed up calculations.

lapply

The function lapply will repeatedly apply a function to each element of a list or vector.

Note that you only specify the name of the function to be called as the second argument. You don't give the function any arguments, unless extra arguments are needed by the function, in which case they are supplied to lapply, not the function.

```
> mean.n.rnorm <- function(n) return(mean(rnorm(n)))
>
> ns <- c(1, 10, 100, 1000)
>
> lapply(ns, mean.n.rnorm)
[[1]]
[1] -0.01890732

[[2]]
[1] 0.1327366

[[3]]
[1] -0.1007226

[[4]]
[1] -0.03226481
>
```

sapply

I usually use sapply, as it returns a vector rather than a list, which I usually find more-convenient to use.

Note that if the function you are calling using one of the *apply functions takes more than one argument, the extra arguments can be listed in the *apply function after the name of the function.

```
>
> ns <- c(1, 10, 100, 1000)
>
> random.nums <- lapply(ns, rnorm)
>
> sapply(random.nums, mean)
[1] 0.34134897 0.30397851 0.03475841 -0.01775784
>
> sapply(random.nums, sd, na.rm = TRUE)
[1] NA 0.9419855 0.8996367 1.0172488
>
```

apply

The apply function is used on matrices and arrays. It applies a function to the rows (MARGIN = 1), or columns (MARGIN = 2) of an array.

```
>
> A <- matrix(1:9, nrow = 3)
>
> A
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
>
> apply(A, MARGIN = 1, max) # max of each row
[1] 7 8 9
>
> apply(A, MARGIN = 2, max) # max of each column
[1] 3 6 9
>
```

apply, continued

The apply function can also be applied to each element, using MARGIN $=$ 1:2.

If you have more than 2 dimensions then those dimensions can also be specified as an argument to MARGIN.

```
>
> A <- matrix(1:9, nrow = 3)
>
> A
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
>
>
> apply(A, MARGIN = 1:2, function(x) return(x**2))
     [,1] [,2] [,3]
[1,]    1   16   49
[2,]    4   25   64
[3,]    9   36   81
>
```

Enough to get started

- Today's class went over the very beginning concepts needed to do statistics.
- Almost all the commands are built into R already. If they aren't in the base R installation, they exist in a separate package.
- The null hypothesis is the "normal" situation, the situation where nothing has changed.
- Hypothesis testing always assumes that the null hypothesis is true.
- Tests determine the likelihood, given the data, of committing a Type 1 error.