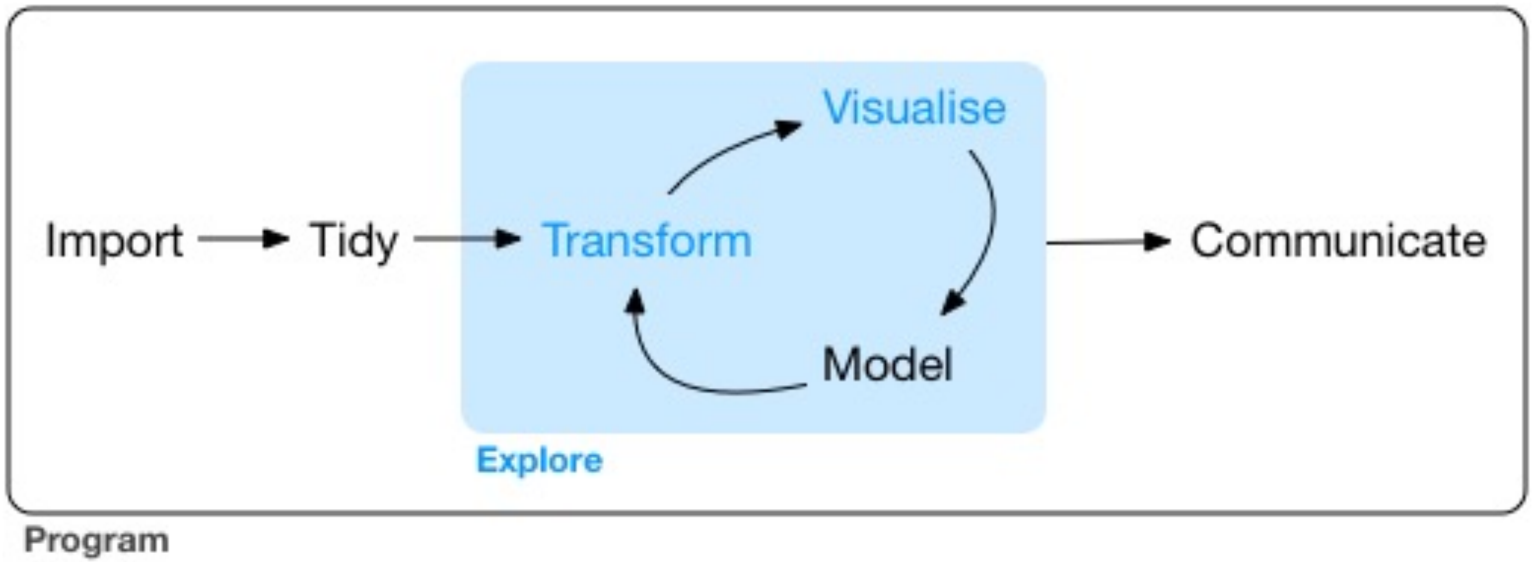
Introduction/Refresher

January 19, 2022

Data science workflow

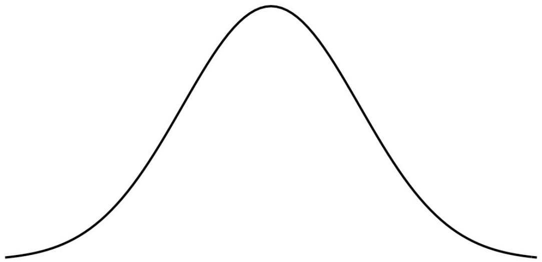


Wickham and Grolemund rhds.had.co.nz

Probability and statistics

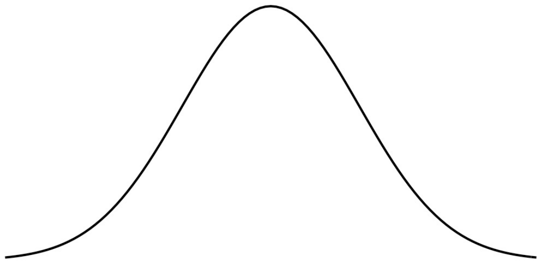
- "Data science" concept takes emphasis away from 1-to-1 mapping between what kind of data you have and what statistical test you choose, and incorporates other vital steps into the process
- Fundamentally, everything can be reduced to identifying probability models that are consistent with your data
- "Statistics" are assigning probabilities (or likelihoods) to different generating models

Common probability distributions



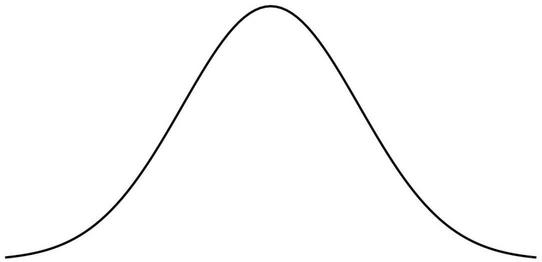
- Normal (Gaussian) "everything is normal"
  - <u>central limit theorem</u>: sum of *independent, identically distributed random variables* tends towards a *normal distribution*
  - very useful for applications because many processes end up being normally distributed even if they don't start that way

Common probability distributions



- Normal (Gaussian) "everything is normal"
  - central limit theorem: sum of *independent, identically distributed random variables* tends towards a *normal distribution*
  - very useful for applications because many processes end up being normally distributed even if they don't start that way

Common probability distributions



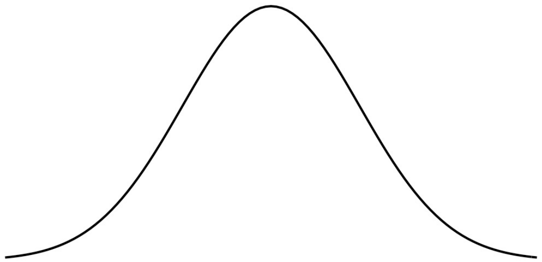
- Normal (Gaussian) "everything is normal"
  - central limit theorem: sum of *independent, identically distributed random variables* tends towards a *normal distribution*
  - very useful for applications because many processes end up being normally distributed even if they don't start that way

independent: the occurrence of one event does not affect the probability of an occurrence of another event

P(A and B) = P(A) × P(B)

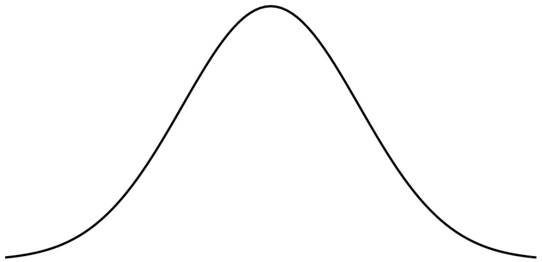
P(A | B) = P(A)   P(B | A) = P(B)

Common probability distributions



- Normal (Gaussian) "everything is normal"
  - <u>central limit theorem</u>: sum of *independent, identically distributed random variables* tends towards a *normal distribution*
  - very useful for applications because many processes end up being normally distributed even if they don't start that way

Common probability distributions

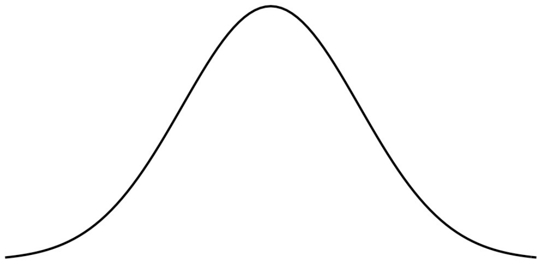


- Normal (Gaussian) "everything is normal"
  - central limit theorem: sum of *independent, identically distributed random variables* tends towards a *normal distribution*
  - very useful for applications because many processes end up being normally distributed even if they don't start that way

random variable: a variable whose value depends on the outcome of a random process

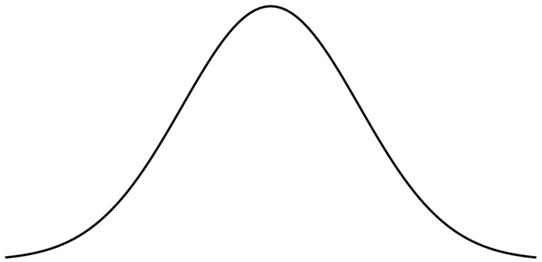
may be discrete (limited number of values) or continuous (any real number)

Common probability distributions



- Normal (Gaussian) "everything is normal"
  - <u>central limit theorem</u>: sum of *independent, identically distributed random variables* tends towards a *normal distribution*
  - very useful for applications because many processes end up being normally distributed even if they don't start that way

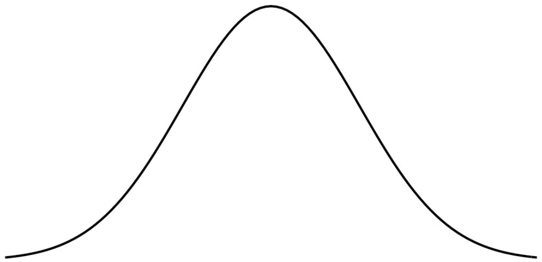
Common probability distributions



- Normal (Gaussian) "everything is normal"
  - central limit theorem: sum of *independent*, *identically distributed random variables* tends towards a *normal distribution*
  - very useful for applications because many processes end up being normally distributed even if they don't start that way

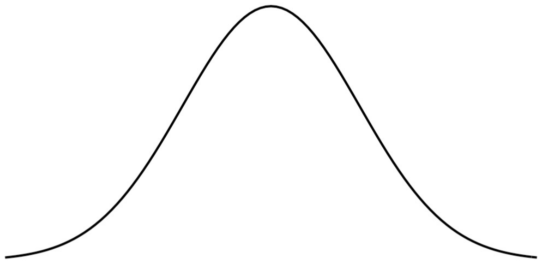
identically distributed: each random variable being summed has the same *distribution*: function that describes the probability of it assuming certain values (or ranges of values)

Common probability distributions



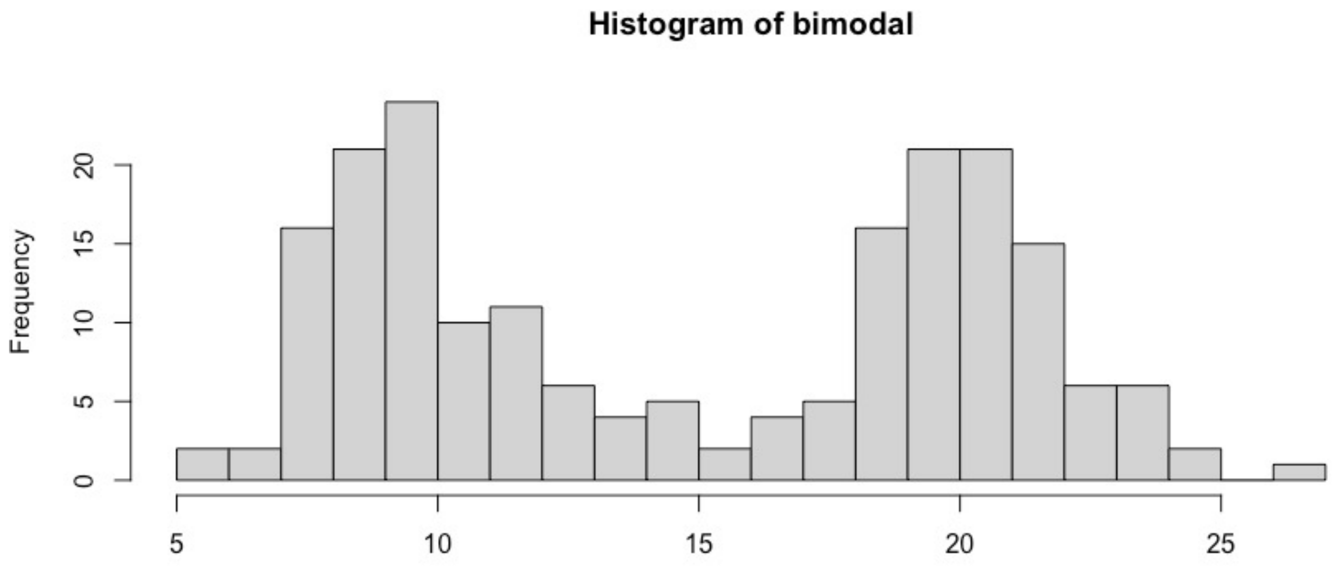
- Normal (Gaussian) "everything is normal"
  - central limit theorem: sum of *independent, identically distributed random variables* tends towards a *normal distribution*
  - very useful for applications because many processes end up being normally distributed even if they don't start that way

Common probability distributions

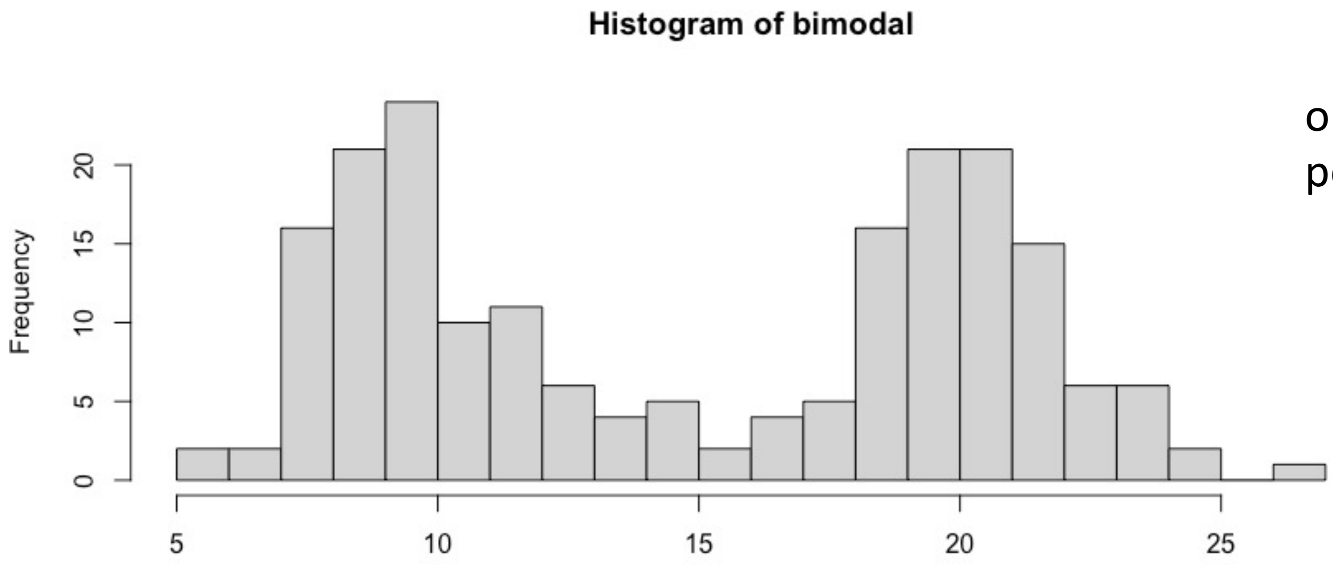


- Normal (Gaussian) "everything is normal"
  - <u>central limit theorem</u>: sum of *independent, identically distributed random variables* tends towards a *normal distribution*
  - very useful for applications because many processes end up being normally distributed even if they don't start that way

normal distribution: classic "bell curve" shaped continuous probability distribution function; provides probabilities that values of a normally distributed random variable will be within a certain interval

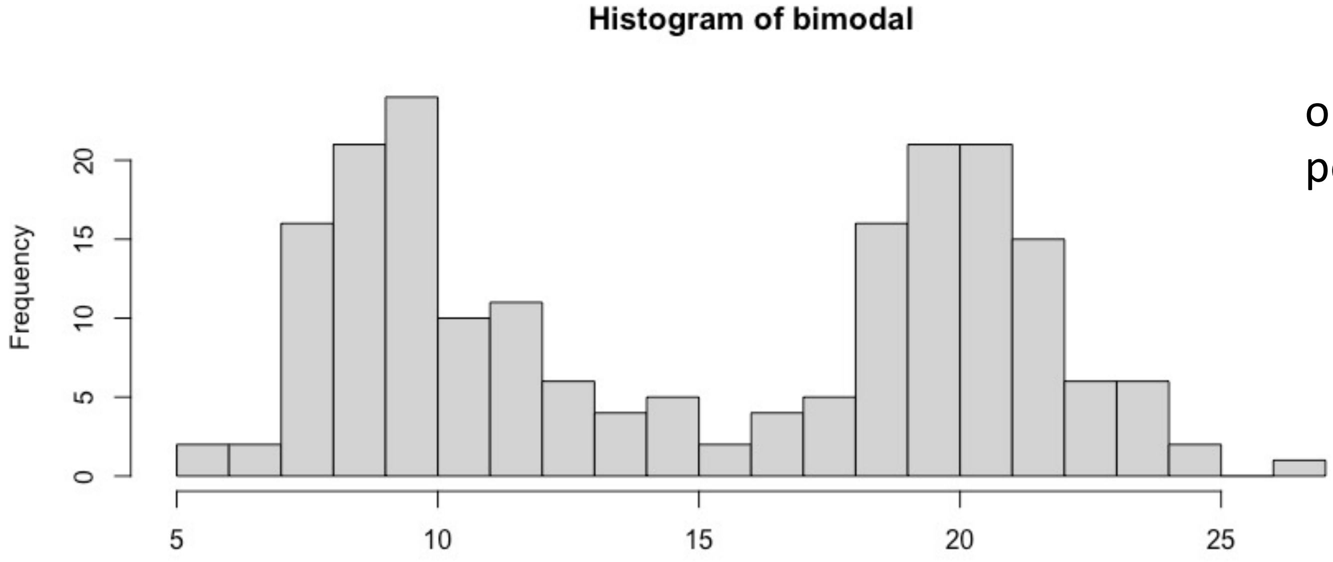


bimodal.R

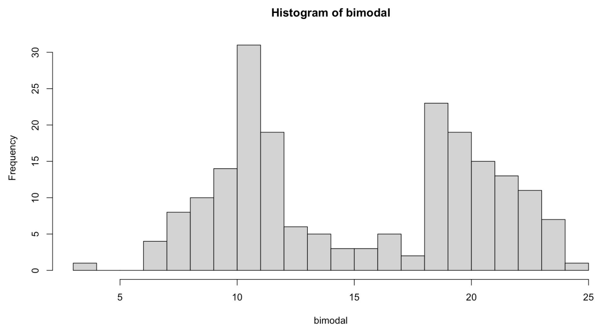


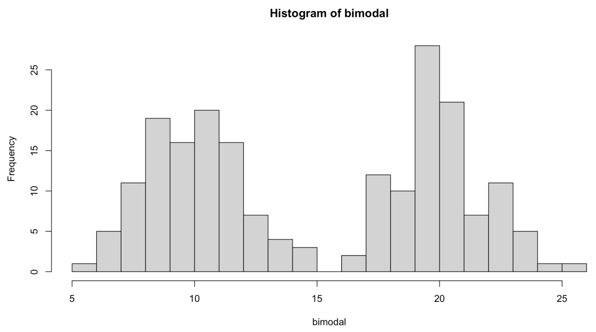
one sample – clearly has two peaks in likely values

bimodal.R

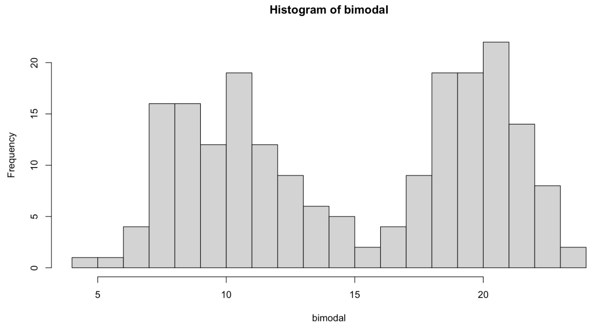


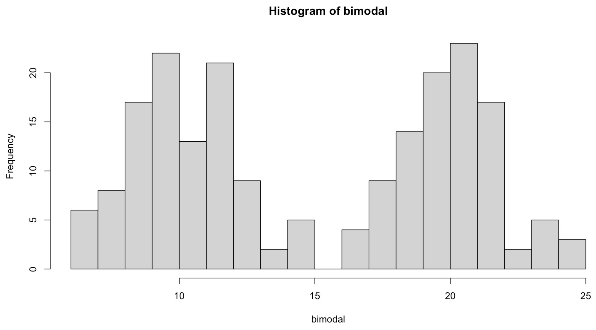
one sample – clearly has two peaks in likely values



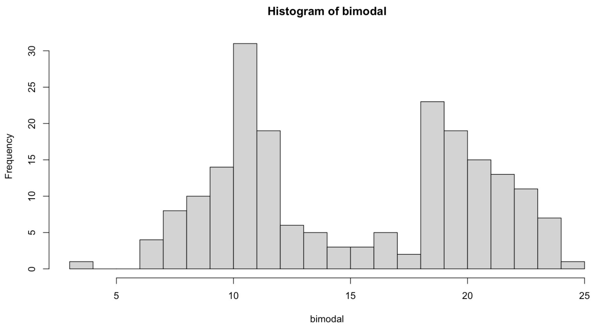


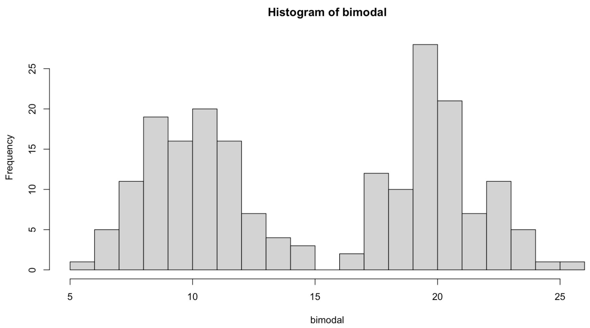
generate other samples from same experiment; different specific values but similar general shapes

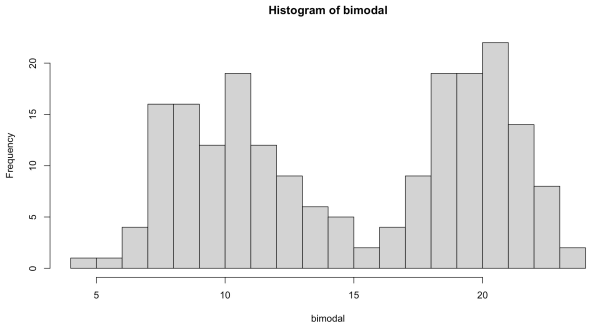


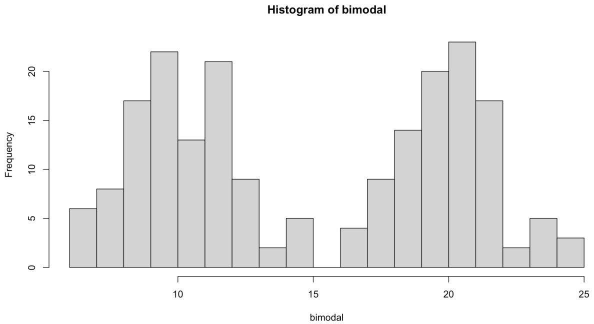


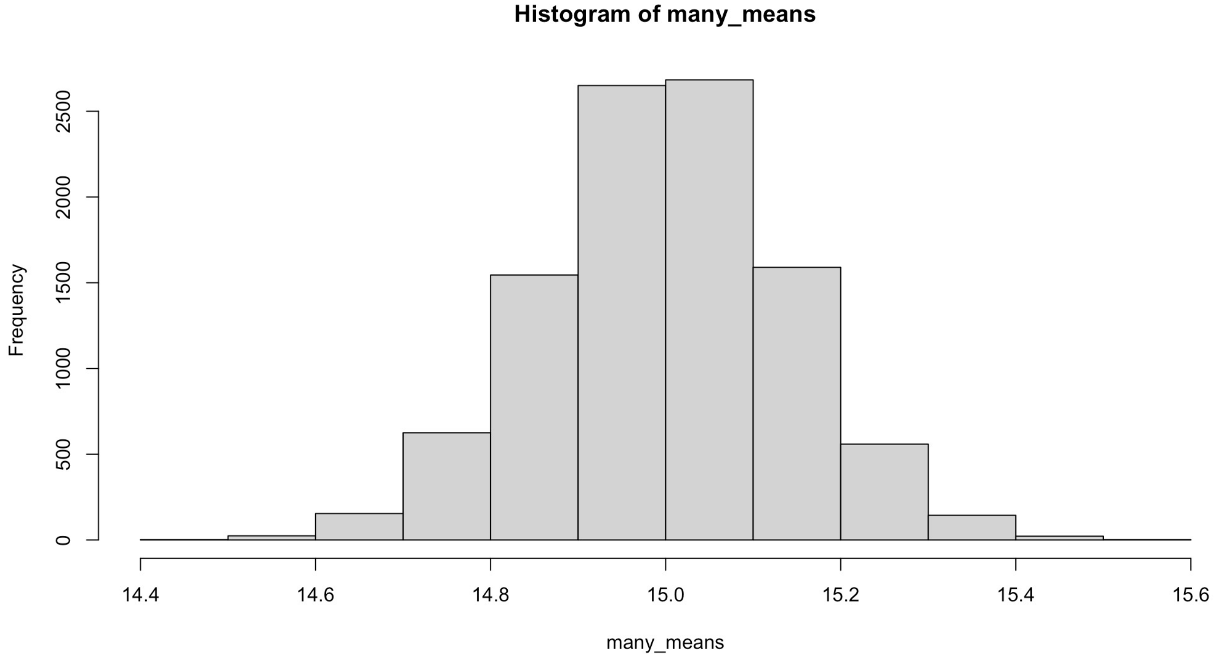
bimodal.R





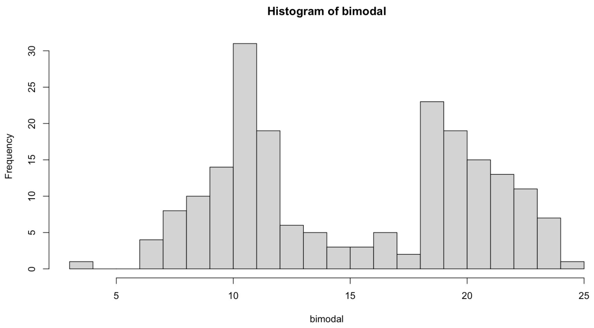


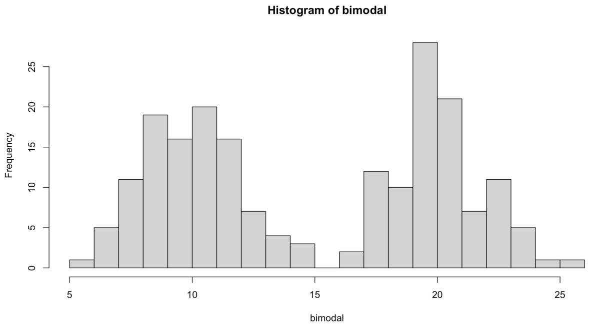


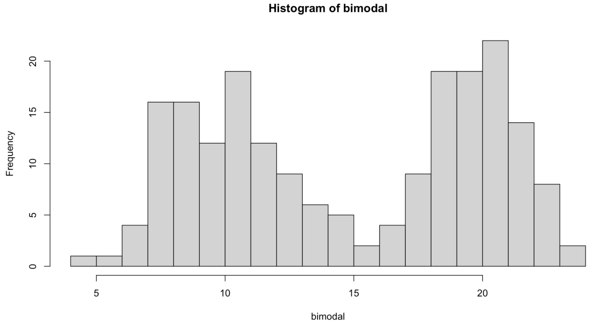


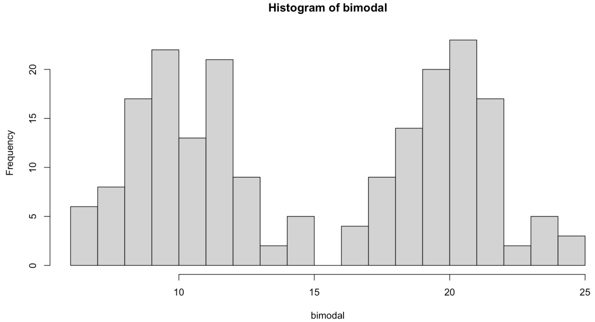
do this many times, take mean from each experiment, plot them: get symmetrical distribution centered on 15

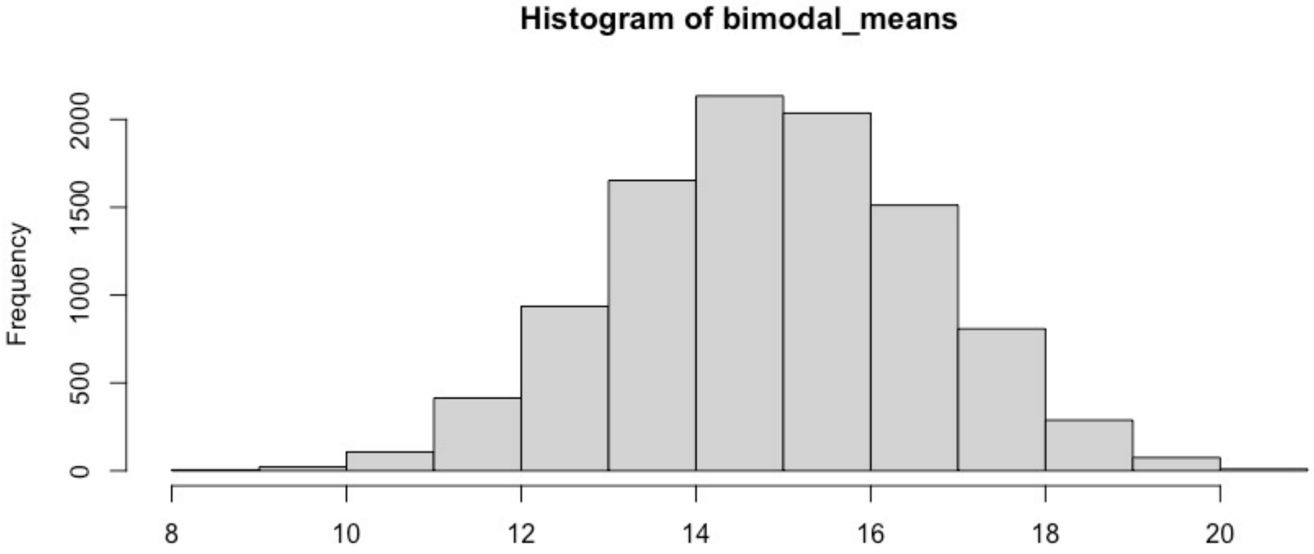
bimodal.R











means of a sample of N = 10 from our original distribution

do this many times, take mean from each experiment, plot them: get (relatively) symmetrical distribution centered on ~ 15

bimodal.R

Common probability distributions

- Normal (Gaussian) "everything is normal"
  - central limit theorem: sum of independent, identically distributed random variables tends towards a normal distribution
  - very useful for applications because many processes end up being normally distributed even if they don't start that way

- Uniform (flat)
- Poisson – counts, "rare event" process, time to first event / failure
- Binomial – coin flips, sums of discrete trials with constant probability

- distributions needn't be symmetrical!

Distributions defined by their parameters

	parameters	mean	variance
normal	mean (μ), standard deviation (σ)	μ	σ^2
uniform	minimum, maximum (a,b)	(a+b)/2	$(b-a)^2/12$
Poisson	lambda (ƛ)	ƛ	ƛ
binomial	probability of "success" (p), number of events/trials (n)	np	np(1-p)

mean = $\mu = \dfrac{\text{sum of the terms}}{\text{number of terms}} = \dfrac{1}{n}\sum_{i=1}^{n} x_i$

variance = $\sigma^2 = \dfrac{\text{sum of squared deviations from mean}}{\text{number of terms}} = \dfrac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$

Easy to simulate probability distributions in R

Normal {stats} R Documentation

The Normal Distribution

Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`.

Usage

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.

Details

If `mean` or `sd` are not specified they assume the default values of `0` and `1`, respectively.

The normal distribution has density

f(x) = 1/(√(2 π) σ) e^-((x - μ)^2/(2 σ^2))

where μ is the mean of the distribution and σ the standard deviation.

?rnorm gives documentation in R

for many different distributions

runif "r unif"
rpois
rbinom

Easy to simulate probability distributions in R

Normal {stats} R Documentation

The Normal Distribution

Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`.

Usage

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.

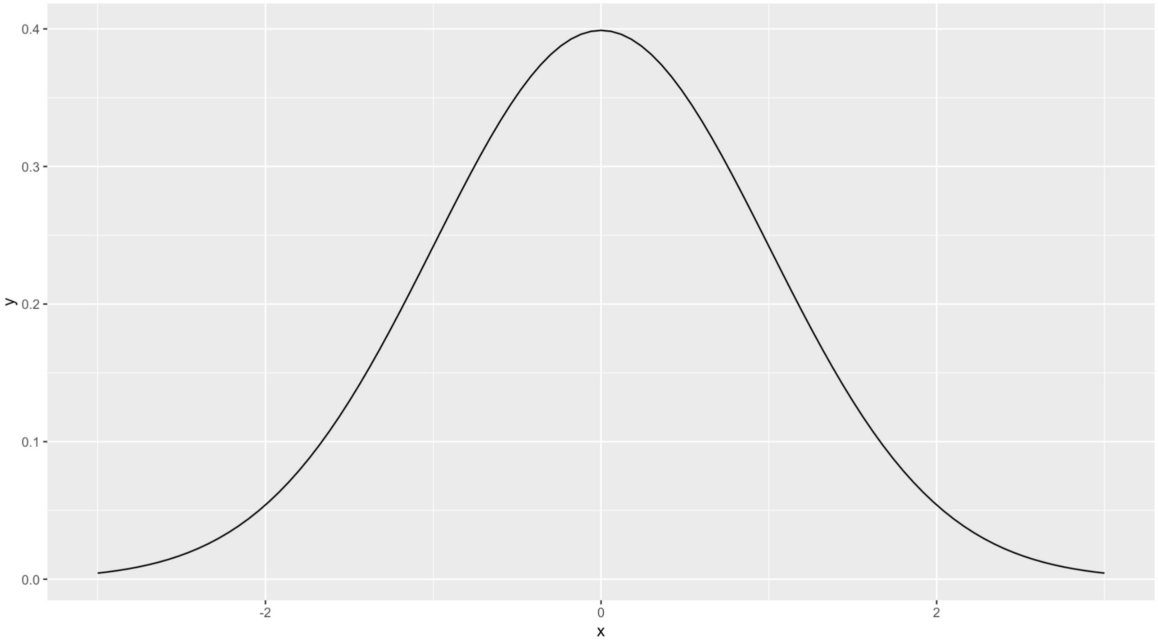
Details

If `mean` or `sd` are not specified they assume the default values of `0` and `1`, respectively.

The normal distribution has density

$$f(x) = 1/(\sqrt{(2 \pi)}\ \sigma)\ e^{\wedge}-((x - \mu)^{\wedge}2/(2\ \sigma^{\wedge}2))$$

where μ is the mean of the distribution and σ the standard deviation.



density for normal with mean 0 and SD 1

Easy to simulate probability distributions in R

Normal {stats} R Documentation

The Normal Distribution

Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`.

Usage

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.

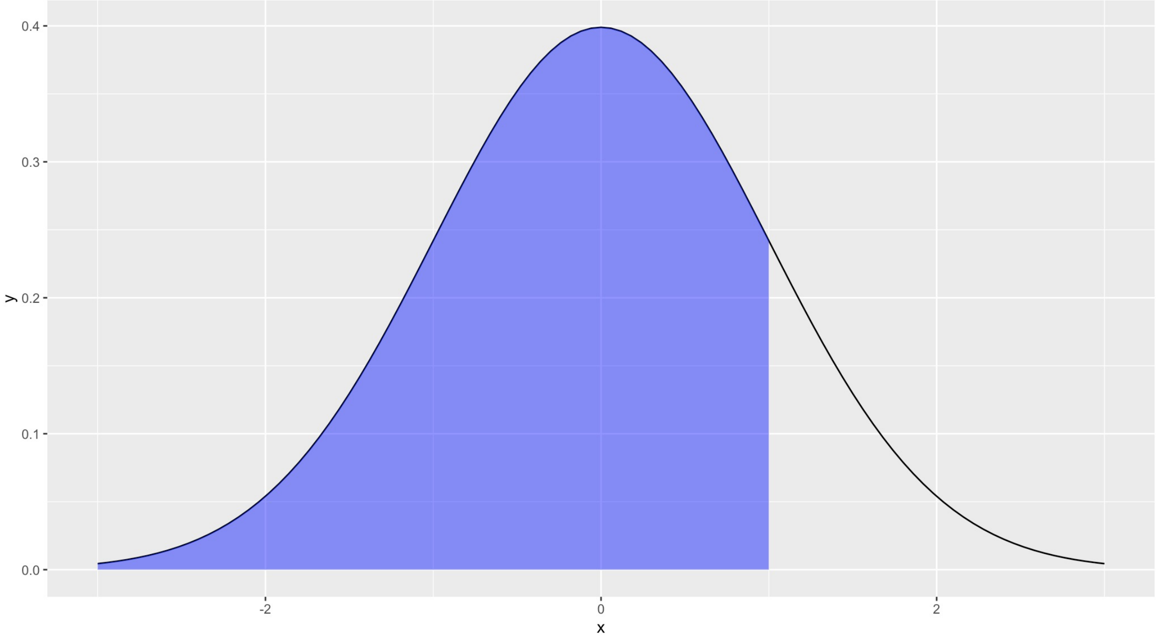
Details

If `mean` or `sd` are not specified they assume the default values of `0` and `1`, respectively.

The normal distribution has density

$$f(x) = 1/(\sqrt{(2 \pi)}\ \sigma)\ e^{\wedge}-((x - \mu)^{\wedge}2/(2\ \sigma^{\wedge}2))$$

where μ is the mean of the distribution and σ the standard deviation.



density for normal with mean 0 and SD 1

pnorm(1) = area under curve less than 1 (~84.1%)

qnorm(.841) = "critical value" of density where area to left is .841 (returns ~ 1)

What kind of data do you have?

- Measurement scale
  - Nominal: categories with no ordering – hair color, favorite condiment
  - Ordinal: categories with ordering – garment size (S/M/L/XL), level of comfort with certain procedures in R
  - Interval: numerical data where there is order and the difference between 2 values is meaningful – temperature in Celsius, GRE scores, pH
  - Ratio: interval data where there is an absolute zero point so the *ratio* between 2 values is also meaningful – temperature in Kelvin, age, height

What kind of data do you have?

- Measurement scale affects how you treat your data
  - Summary statistics: cannot have a mean of "eye color" – frequency of different categories may make more sense (for example)
  - <u>Pay attention to how R codes variables when you read in data</u>: character / factor vs numeric
  - R does not know that mouse 2 is not twice as much as mouse 1
  - Values of 1, 2, 3, 4 behave differently in analyses depending on how they are represented
  - We will have more to say about this for specific analyses but it's always a good idea to make sure your data are read into R in a sensible

What kind of data do you have?

- Consider how your variables are measured
- Reliability - how consistently does the variable measure what it is intended to measure
  - May include but not limited to aspects of precision (how precisely can you measure a physical quantity, how much noise in measurement)
  - Test-retest reliability (if measurement is stable over time); interrater reliability
- Validity – how accurately does variable measure construct of interest
  - "Face" validity – does it seem like it relates to the concept
  - "Construct" validity – do measures intended to capture the same concept relate to one another ("discriminant" validity – don't relate to distinct concepts)

Statistical tests and power

- Get your data, do a test, evaluate the p value
- What you are doing is evaluating the probability of observing an outcome as extreme or more extreme in your data under a "null hypothesis" of no effect
- If this is sufficiently unlikely under the null hypothesis, you "reject" the null hypothesis
- A 5% probability under the null is conventional ($p < .05$) to talk about a finding as if it is "true"

TRUTH

Your statistical test

	No effect in population (H_o true)	True effect in population (H_o false)
$p > .05$ Test not significant	😀	Type II error (β)
$p < .05$ Test significant	Type I error (α)	😀 Power ($1-\beta$)

Statistical tests and power

- Of course, we do not know what the truth is when we do an experiment

- We will have more to say about power and how to interpret the results of statistical tests

- Tradeoff between significance level, power, sample size

What are you actually doing when you do a statistical test?

- "p-value" is the probability of observing results as extreme (or more extreme) than yours <u>under the assumptions of the null hypothesis</u>
  - these assumptions may include that the data are distributed a certain way, that certain independence conditions are met, ...
  - if these assumptions do not hold, the p-value is probably wrong

- this probability may be determined a number of ways
  - analytically (calculate probabilities of all possible outcomes)
  - via simulations (e.g. permutation test)
  - using probability distributions (e.g. normal distribution)

Common statistical tests

- "parametric" vs "nonparametric"
    - assumption about underlying probability distribution of data, **independence**
    - as a general rule, parametric tests are more powerful but involve making some assumptions about the distribution of your data or about the probability model you are using
    - nonparametric tests typically involve removing information about magnitude and scale and just focus on order (ranks)

- Measurement scale affects your choice of statistical test
    - "flow chart" of picking tests
    - choice of underlying probability model to represent data

- "general linear model"
    - rather than laundry list of tests, many tests can be conceptualized as special cases of a very general statistical model

Common statistical tests

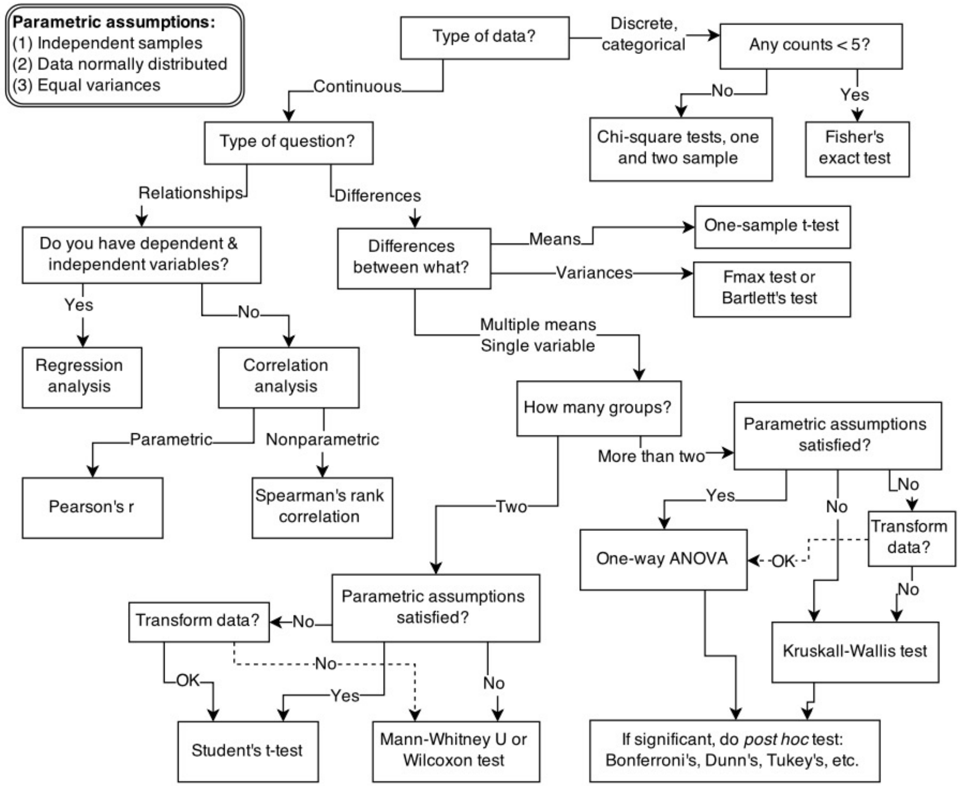


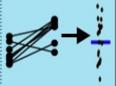
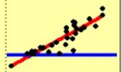
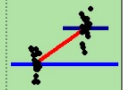
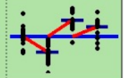
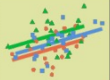
FIGURE 1.1. Example decision tree, or flowchart, for selecting an appropriate statistical procedure. Beginning at the top, the user answers a series of questions about measurement and intent, arriving eventually at the name of a procedure. Many such decision trees are possible.

McElreath (2020) *Statistical Rethinking* (2nd ed)

Common statistical tests are linear models

Last updated: 02 April, 2019

See worked examples and more details at the accompanying notebook: https://lindeloev.github.io/tests-as-linear

	Common name	Built-in function in R	Equivalent linear model in R	Exact?	The linear model in words	Icon
Simple regression: lm(y ~ 1 + x)	**y is independent of x** P: One-sample t-test N: Wilcoxon signed-rank	t.test(y) wilcox.test(y)	lm(y ~ 1) lm(signed_rank(y) ~ 1)	✓ for N >14	One number (intercept, i.e., the mean) predicts **y**. - (Same, but it predicts the *signed rank* of **y**.)	
	P: Paired-sample t-test N: Wilcoxon matched pairs	t.test(y_1, y_2, paired=TRUE) wilcox.test(y_1, y_2, paired=TRUE)	lm(y_2 - y_1 ~ 1) lm(signed_rank(y_2 - y_1) ~ 1)	✓ for N >14	One intercept predicts the pairwise **y_2-y_1** differences. - (Same, but it predicts the *signed rank* of **y_2-y_1**.)	
	y ~ continuous x P: Pearson correlation N: Spearman correlation	cor.test(x, y, method='Pearson') cor.test(x, y, method='Spearman')	lm(y ~ 1 + x) lm(rank(y) ~ 1 + rank(x))	✓ for N >10	One intercept plus **x** multiplied by a number (slope) predicts **y**. - (Same, but with *ranked* **x** and **y**)	
	y ~ discrete x P: Two-sample t-test P: Welch's t-test N: Mann-Whitney U	t.test(y_1, y_2, var.equal=TRUE) t.test(y_1, y_2, var.equal=FALSE) wilcox.test(y_1, y_2)	lm(y ~ 1 + G_2)^A gls(y ~ 1 + G_2, weights=...^{B})^A lm(signed_rank(y) ~ 1 + G_2)^A	✓ ✓ for N >11	An intercept for **group 1** (plus a difference if **group 2**) predicts **y**. - (Same, but with one variance *per group* instead of one common.) - (Same, but it predicts the *signed rank* of **y**.)	
Multiple regression: lm(y ~ 1 + x_1 + x_2 +...)	P: One-way ANOVA N: Kruskal-Wallis	aov(y ~ group) kruskal.test(y ~ group)	lm(y ~ 1 + G_2 + G_3 +...+ G_N)^A lm(rank(y) ~ 1 + G_2 + G_3 +...+ G_N)^A	✓ for N >11	An intercept for **group 1** (plus a difference if group ≠ 1) predicts **y**. - (Same, but it predicts the *rank* of **y**.)	
	P: One-way ANCOVA	aov(y ~ group + x)	lm(y ~ 1 + G_2 + G_3 +...+ G_N + x)^A	✓	- (Same, but plus a slope on **x**.) *Note: this is discrete AND continuous. ANCOVAs are ANOVAs with a continuous x.*	
	P: Two-way ANOVA	aov(y ~ group * sex)	lm(y ~ 1 + G_2 + G_3 + ... + G_N + S_2 + S_3+ ... + S_K + G_2*S_2+G_3*S_3+...+G_N*S_K)	✓	Interaction term: changing **sex** changes the **y ~ group** parameters. *Note: **$G_{2 \text{ to } N}$** is an indicator (0 or 1) for each non-intercept levels of the **group** variable. Similarly for **$S_{2 \text{ to } K}$** for sex. The first line (with G_i) is main effect of group, the second (with S_i) for sex and the third is the **group × sex** interaction. For two levels (e.g. male/female), line 2 would just be "S_2" and line 3 would be S_2 multiplied with each G_i.*	[Coming]
	Counts ~ discrete x N: Chi-square test	chisq.test(groupXsex_table)	**Equivalent log-linear model** glm(y ~ 1 + G_2 + G_3 + ... + G_N + S_2 + S_3+ ... + S_K + G_2*S_2+G_3*S_3+...+G_N*S_K, family=...)^A	✓	Interaction term: (Same as Two-way ANOVA.) *Note: Run glm using the following arguments:* `glm(model, family=poisson())` *As linear-model, the Chi-square test is $\log(y_i) = \log(N) + \log(\alpha_i) + \log(\beta_j) + \log(\alpha_i\beta_j)$ where α_i and β_j are proportions. See more info in the accompanying notebook.*	*Same as Two-way ANOVA*
	N: Goodness of fit	chisq.test(y)	glm(y ~ 1 + G_2 + G_3 +...+ G_N, family=...)^A	✓	(Same as One-way ANOVA and see Chi-Square note.)	*1W-ANOVA*

List of common parametric (P) non-parametric (N) tests and equivalent linear models. The notation y ~ 1 + x is R shorthand for y = 1·b + a·x which most of us learned in school. Models in similar colors are highly similar, but really, notice how similar they *all* are across colors! For non-parametric models, the linear models are reasonable approximations for non-small sample sizes (see "Exact" column and click links to see simulations). Other less accurate approximations exist, e.g., Wilcoxon for the sign test and Goodness-of-fit for the binomial test. The signed rank function is `signed_rank = function(x) sign(x) * rank(abs(x))`. The variables G_i and S_i are "dummy coded" indicator variables (either 0 or 1) exploiting the fact that when Δx = 1 between categories the difference equals the slope. Subscripts (e.g., G_2 or y_1) indicate different columns in data. lm requires long-format data for all non-continuous models. All of this is exposed in greater detail and worked examples at https://lindeloev.github.io/tests-as-linear.

^{A} See the note to the two-way ANOVA for explanation of the notation.
^{B} Same model, but with one variance per group: `gls(value ~ 1 + G2, weights = varIdent(form = ~1|group), method="ML")`.

Jonas Kristoffer Lindeløv
https://lindeloev.net

https://lindeloev.github.io/tests-as-linear

What will we focus on?

- Power analysis: what are we hoping to achieve with our experiment and what do we expect? Includes concept of effect size
- Regression / analysis of variance: simple case of general linear model
- Multilevel models ("linear mixed models"): more complex case where we need to model dependency in data points
- Time series analyses: generating process for data involves time-varying periodic components
- "Data mining": clustering, dimensionality reduction – exploring structure in data

What are Mark and Erin's goals?

- Show you a variety of statistical and computational methods, so that even if they don't apply to your research, you can understand them better when you encounter them in the literature.
- Bridge the gap between theory and practice in statistics and data analysis. Real data are messy and you get away from "textbook" analysis cases very quickly.
- Help you anticipate issues you may encounter in preparing fellowship applications.
- Serve as resources for your research.

General principles

- Reproducible code (be able to reconstruct how you got your analytic results)
- Good data management practices (don't edit your raw data, document everything)
- Visualization: choice of good graphics for particular kinds of data to help the viewer make correct inferences
- Use of simulations to explore data analysis problems

Why R?

- Good for data analysis – many sophisticated statistical procedures implemented in R
- Easily reproducible – run analyses by writing code vs. point and click
- Develop workflow that streamlines steps in data science process
- **Open source / free**

- Learning curve is steep
- Not ideal for data entry (requires different tools)
- Who wrote the package you're using?
- Not always ideal for quick result (cf. `jamovi`)

Grading

- Letter graded courses are better for fellowship applications because for some stupid reason reviewers pay attention to grad school grades.

- Our goal with homework assignments is that you try things for yourself and work through problems, perhaps running into unexpected issues (like what you'd have analyzing real data)

- We will often provide interactive feedback on homework on Slack to help you get the most out of it.