Frequency evaluations

- Bayesian theory has epistemic and aleatory probabilities
- Frequency evaluations focus on frequency properties given aleatoric repetition of an observation and modeling
  - Asymptotic consistency
  - Unbiasedness
    - not that important in Bayesian inference, small error more important
  - Efficiency
    - small squared error
    - other utility/cost functions possible
  - Calibration
    - α%-posterior interval has the true value in α% cases
    - α%-predictive interval has the true future values in α% cases
    - approximate calibration with shorter intervals for likely true values more important than exact calibration with bad intervals for all possible values.

Frequentist statistics

- Frequentist statistics accepts only aleatory probabilities
  - Estimates are based on data
  - Uncertainty of estimates are based on all possible data sets which could have been generated by the data generating mechanism
    - inference is based also on data we did not observe
- Estimates are derived to fulfill frequency properties
  - Maximum likelihood fulfills just asymptotic frequency properties
  - Common desiderata are 1) unbiasedness, 2) minimum variance, 3) calibration of confidence interval

Frequentist statistics

- Estimates are derived to fulfill frequency properties
  - Maximum likelihood fulfills just asymptotic frequency properties
  - Common desiderata are 1) unbiasedness, 2) minimum variance, 3) calibration of confidence interval
- Requirement of unbiasedness may lead to higher variance or silly estimates
  - unbiased estimate for strictly positive parameter can be negative
- Confidence interval is defined to have true value inside the interval in α% cases of repeated data generation from the data generating mechanism
  - doesn't say how likely the true value is inside the interval given the observed data
  - doesn't need be useful to have perfect calibration

Frequentist vs Bayes vs others

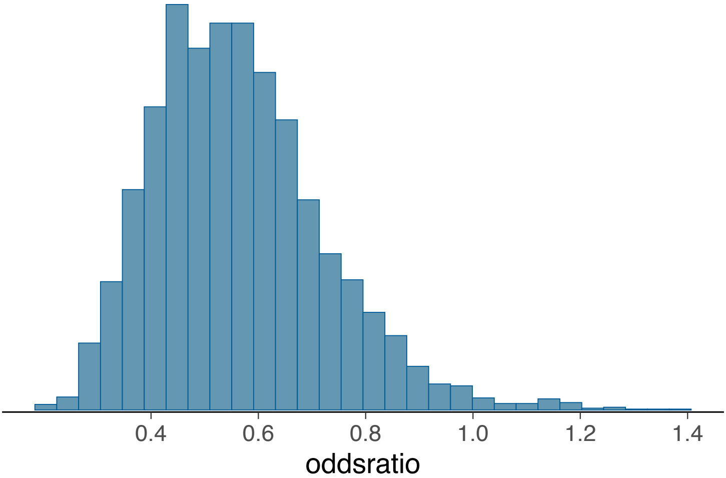
- There is a great amount of very useful frequentist statistics
  - also for simple models and lot's of data there is not much difference
- Bayesian inference
  - easier for complex, e.g. hierarchical, models
  - easier when model changes
  - a consistent way to add prior information
- Lot of machine learning is not pure frequentist or Bayesian

Hypothesis testing

- Frequentist approach can be used to to make estimates and confidence intervals, but for some reason null hypothesis testing has a very big role
  - reporting just the null hypothesis testing result throws away lot of useful information
  - some Bayesians are also into null hypothesis testing
- Frequentist null hypothesis testing
  - asks what if data is generated from the smaller model
  - doesn’t tell whether the more complex model is good enough

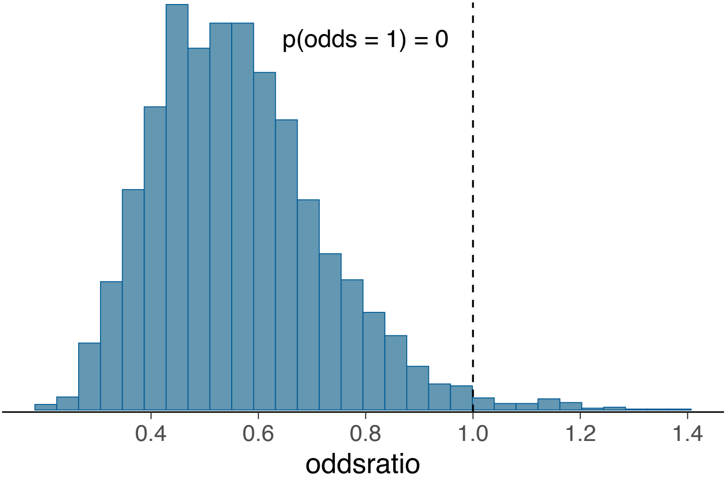
Bayesian hypothesis testing

- Instead of hypothesis testing, report full posterior and
  - compare to expert information
  - combine with utility/cost function



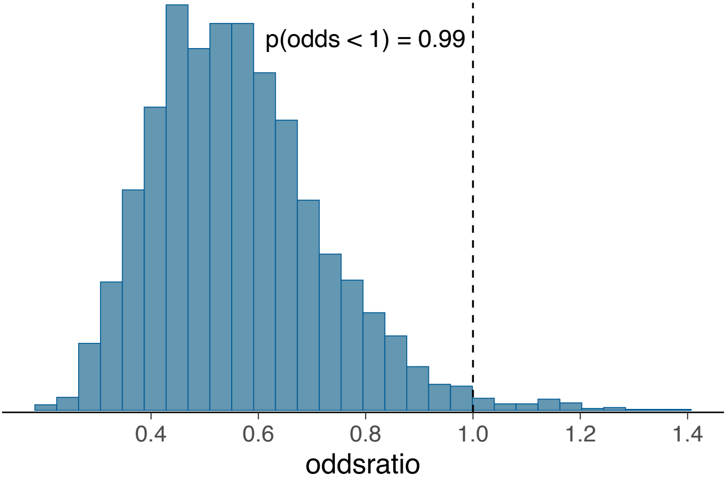
Bayesian hypothesis testing

- Instead of hypothesis testing, report full posterior
  - for continuous posterior there is zero probability that e.g. treatment effect is exactly zero



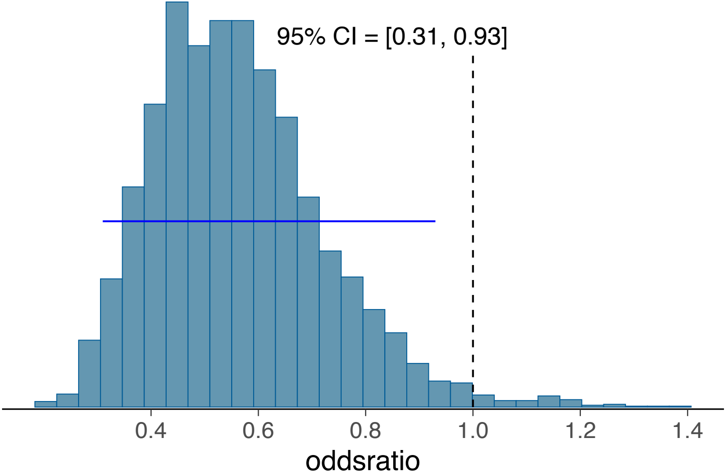
Bayesian hypothesis testing

- Instead of hypothesis testing, report full posterior
  - for continuous posterior we could compute the probability that we know the sign of the effect



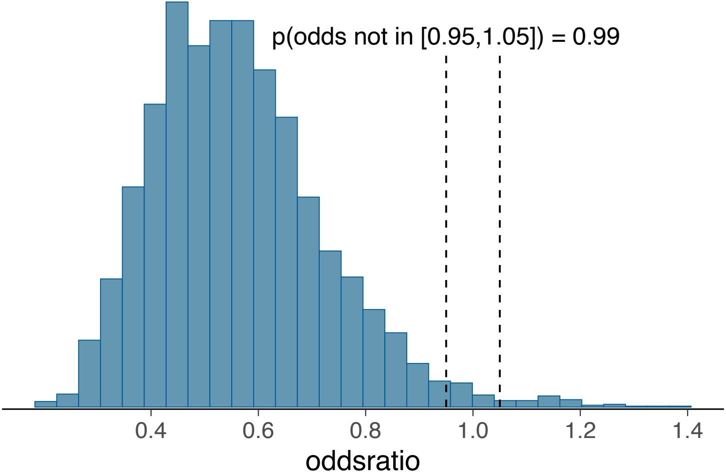
Bayesian hypothesis testing

- Instead of hypothesis testing, report full posterior
  - for continuous posterior some people compare whether posterior interval includes null case



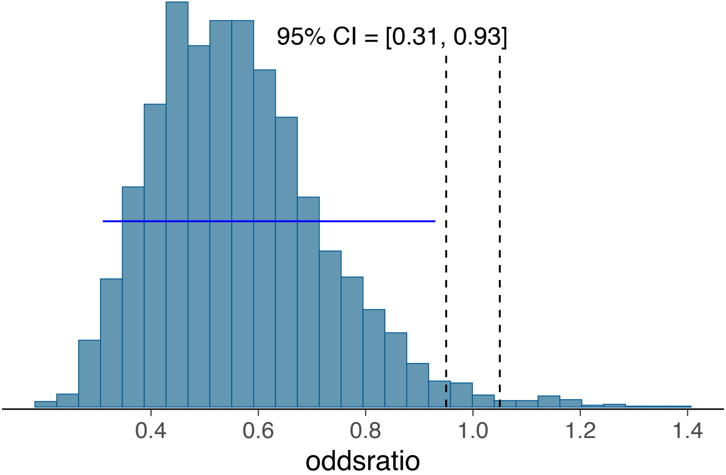
Bayesian hypothesis testing

- Equivalence testing (region of practical equivalence)
  - what is the probability that the effect is closer than ϵ to null, where ϵ is based on what is practically useful effect size



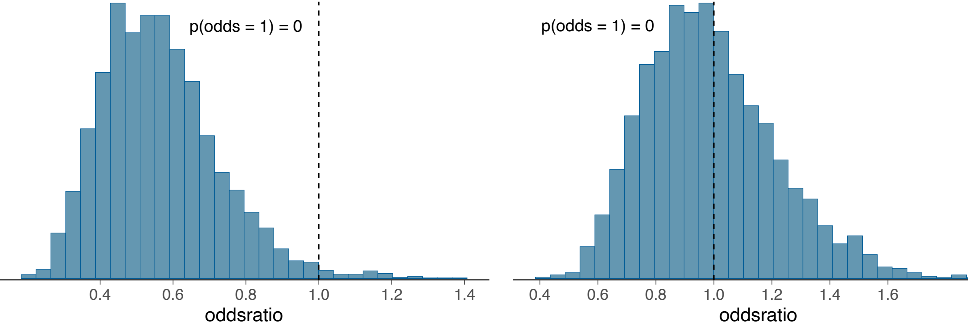
Bayesian hypothesis testing

- Equivalence testing (region of practical equivalence)
  - some people combine posterior interval and region of practical equivalence



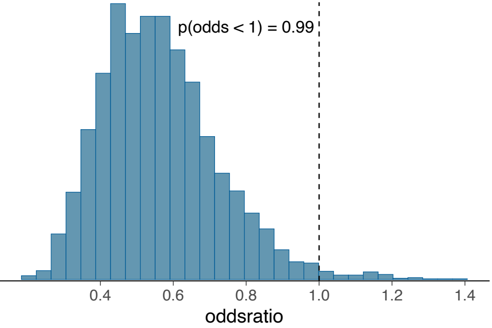
Bayesian hypothesis testing

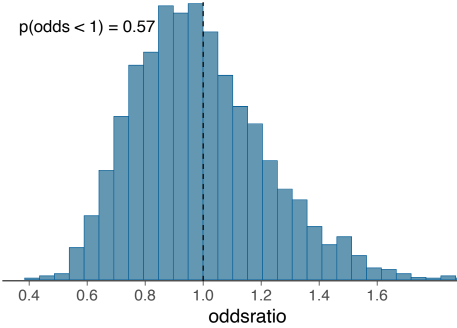
- Instead of hypothesis testing, report full posterior
  - for continuous posterior there is zero probability that e.g. treatment effect is exactly zero



Bayesian hypothesis testing

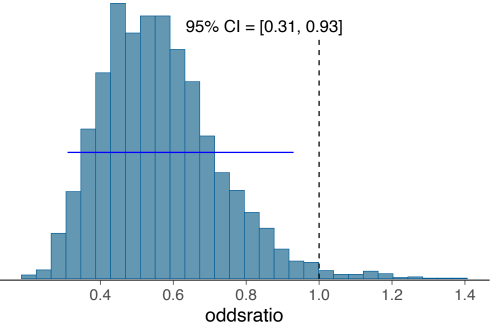
- Instead of hypothesis testing, report full posterior
  - for continuous posterior we could compute the probability that we know the sign of the effect

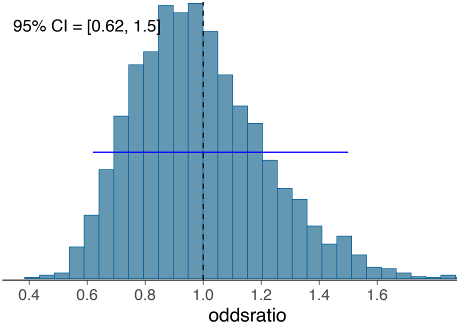




Bayesian hypothesis testing

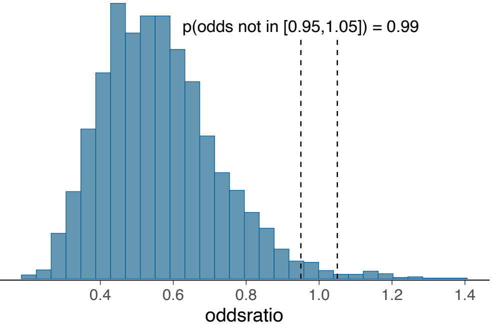
- Instead of hypothesis testing, report full posterior
  - for continuous posterior some people compare whether posterior interval includes null case

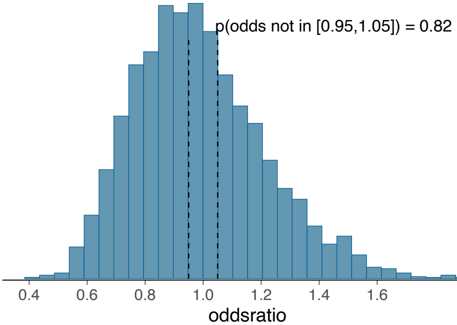




Bayesian hypothesis testing

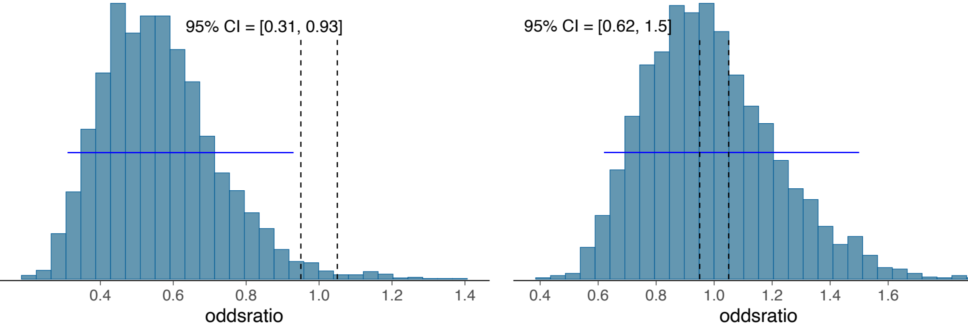
- Instead of hypothesis testing, report full posterior
  - region of practical equivalence (ROPE)





Bayesian hypothesis testing

- Instead of hypothesis testing, report full posterior
  - region of practical equivalence (ROPE)

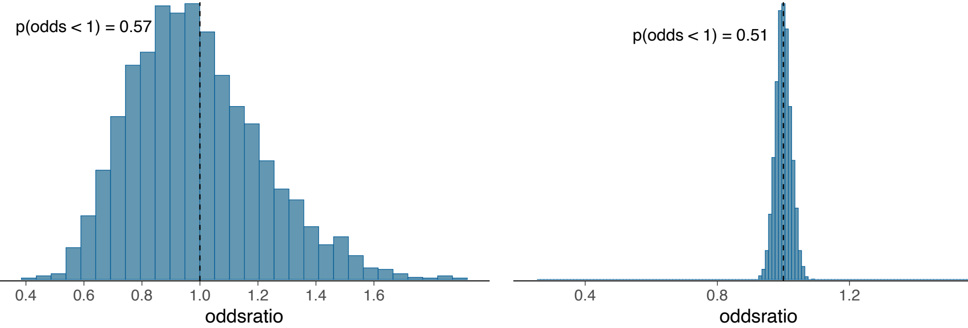


Bayesian hypothesis testing

- Instead of hypothesis testing, report full posterior
  - for continuous posterior there is zero probability that e.g. treatment effect is exactly zero

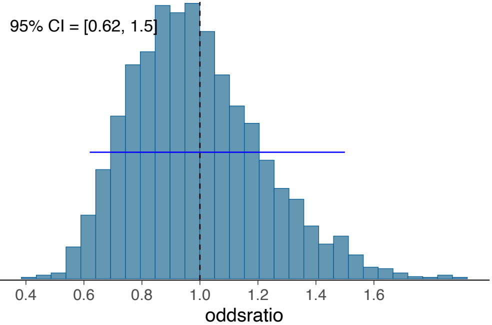
Bayesian hypothesis testing

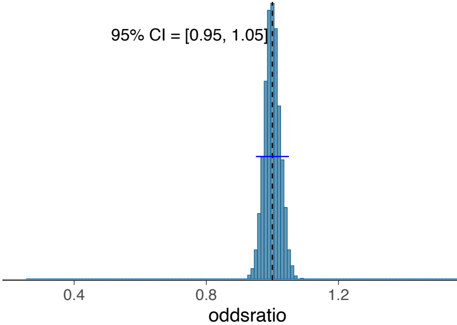
- Instead of hypothesis testing, report full posterior
  - for continuous posterior we could compute the probability that we know the sign of the effect



Bayesian hypothesis testing

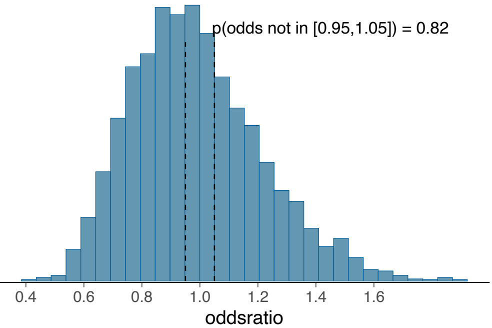
- Instead of hypothesis testing, report full posterior
  - for continuous posterior some people compare whether posterior interval includes null case

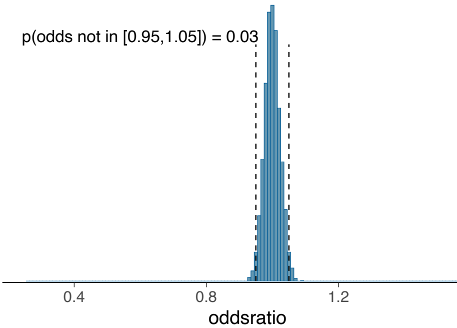




Bayesian hypothesis testing

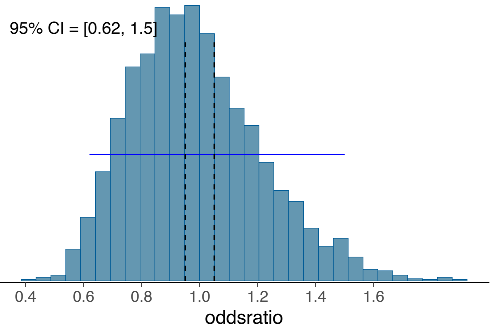
- Instead of hypothesis testing, report full posterior
  - region of practical equivalence (ROPE)

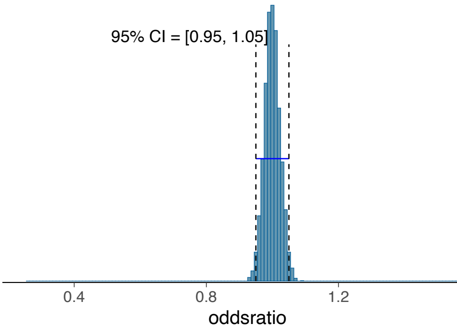




Bayesian hypothesis testing

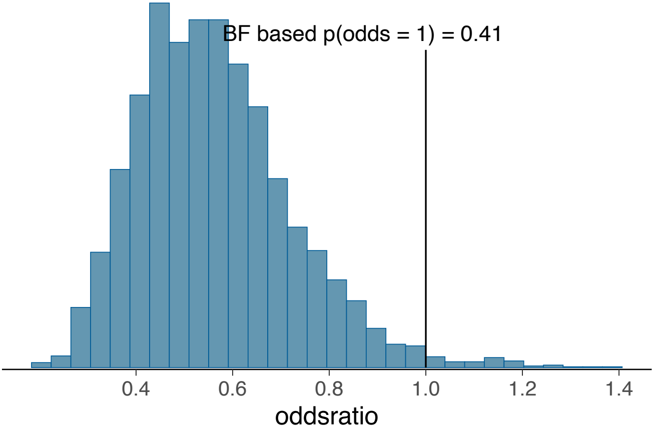
- Instead of hypothesis testing, report full posterior
  - region of practical equivalence (ROPE)





Bayesian hypothesis testing

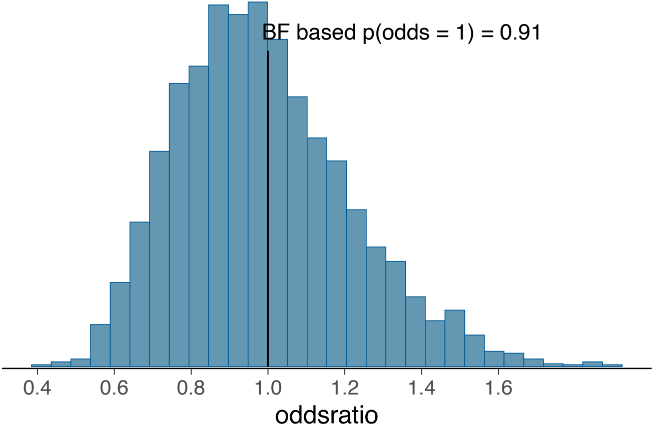
- Bayes factor
  - null model has, e.g., the treatment effect fixed to 0
  - assumes that there is non-zero probability that the treatment effect can be exactly zero
  - requires posterior inference for the null model, too



with `bridgesampling` package, see also BDA3 13.10

Bayesian hypothesis testing

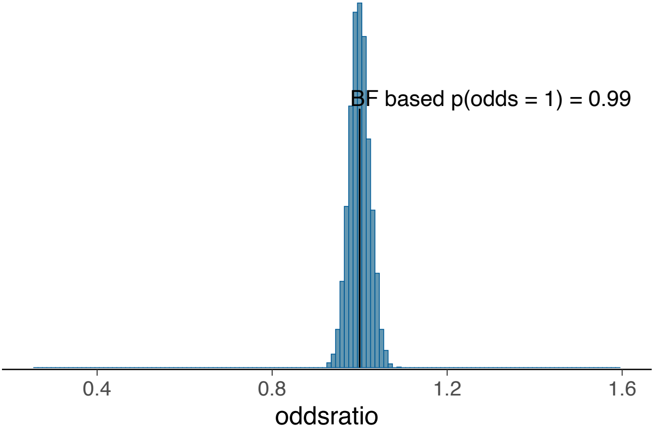
- Bayes factor
  - null model has, e.g., the treatment effect fixed to 0
  - assumes that there is non-zero probability that the treatment effect can be exactly zero
  - requires posterior inference for the null model, too



with `bridgesampling` package, see also BDA3 13.10

Bayesian hypothesis testing

- Bayes factor
  - null model has, e.g., the treatment effect fixed to 0
  - assumes that there is non-zero probability that the treatment effect can be exactly zero
  - requires posterior inference for the null model, too



with `bridgesampling` package, see also BDA3 13.10

Bayesian hypothesis testing

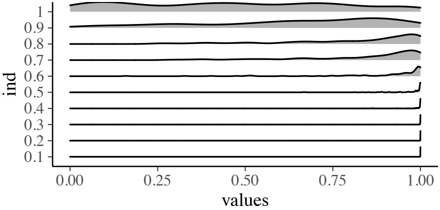
- Predictive performance
  - is there difference in predictive performance with, e.g., treatment effect fixed to zero or unknown treatment effect
  - requires posterior inference for the null model or projection from the full to null
  - looking at the posterior is better if parameters are independent

In the beta blockers example

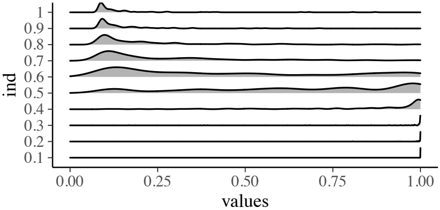
- Leave-one-group-out is not sensible as there are only two groups
- Leave-one-person-out works, but is less efficient than looking at the posterior (see https://avehtari.github.io/modelselection/betablockers.html)

Simulation experiment

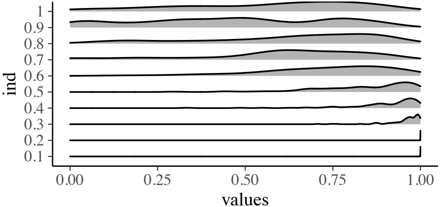
p(odds < 1)



Marginal likelihood comparison



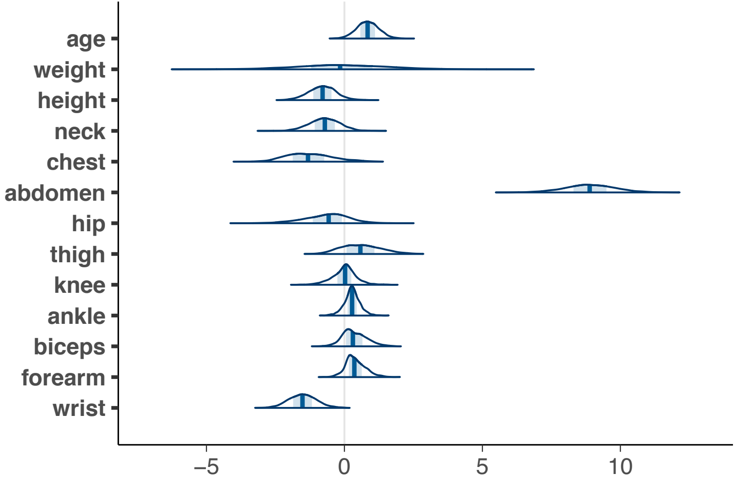
LOO comparison



Hypothesis testing and posterior dependencies

Looking at the marginal posterior(s) can be misleading when there are many parameters

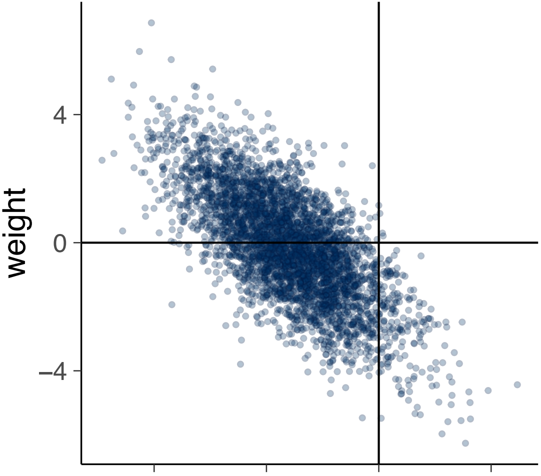
Marginal posteriors of coefficients



Hypothesis testing and posterior dependencies

Looking at the marginal posterior(s) can be misleading when there are many parameters

Bivariate marginal of weight and height



Hypothesis testing and posterior dependencies

In bodyfat example, starting from full model

- BF in favor of removing weight (p=0.92)
- LOO in favor of removing weight (p=0.99)

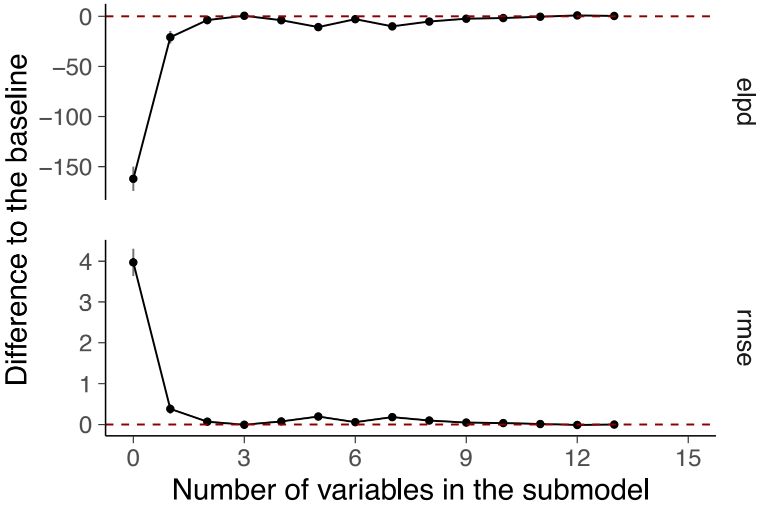
In bodyfat example, starting from model y $\sim$ abdomen

- BF in favor of adding weight (p=1.0)
- LOO in favor of adding weight (p=1.0)

Variable selection

More elaborate approaches are needed for variable selection

See Lecture 9.3 on projection predictive variable selection



Common statistical tests as Bayesian models

Most common statistical tests are linear models

t-test	mean of data	`stan_glm(y ~ 1)`
paired *t*-test	mean of diffs	`stan_glm((y1 - y2) ~ 1)`
Pearson correl.	linear model	`stan_glm(y ~ 1 + x)`
two-sample *t*-test	group means	`stan_glm(y ~ 1 + gid)`
ANOVA	hier. model	`stan_glm(y ~ 1 + (1 | gid))`
...		

possible to extend, e.g., with group specific variances and and different distributions such *t*- or Poisson distribution

See longer list and illustrations (with `lm`) at
https://lindeloev.github.io/tests-as-linear/
and
in the forthcoming *Regression and other stories* book

Chapter 8: Modelling accounting for data collection

Highly recommended to read. Very informative, but also dense chapter.

- We need to model the data collection unless it is ignorable
- We need to know when data collection is ignorable
- Data collection
  - Sample surveys
  - Designed experiments
  - Randomization
  - Observational studies
  - Censoring and truncation

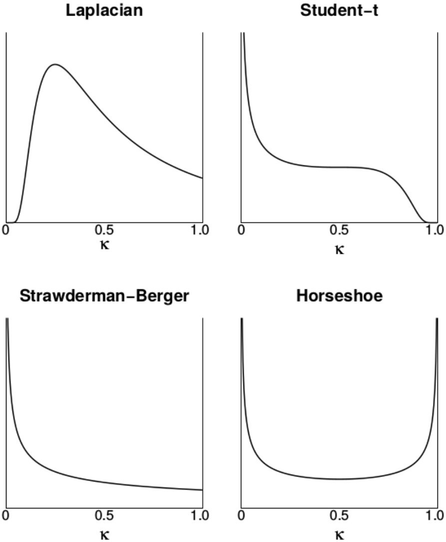
Chapter 14: Introduction to regression models

- Justification of conditional modeling
  - if joint model factorizes $p(y, x|\theta, \phi) = p(y|x, \theta)p(x|\phi)$ we can model just $p(y|x, \theta)$
- Gaussian linear model with conjugate prior
  - the conditional posterior is multivariate normal
  - with fixed prior on weights, the joint posterior is N-Inv-χ^2
  - these properties are sometimes useful and thus good to know, but with probabilistic programming less often needed
- Bit on causal analysis (see much more in ROS)
- Assembling matrix of explanatory variables
  - identifiability, collinearity, nonlinear relations, indicator and categorical variables, interactions
  - variable selection is not much discussed (see lectures 9.2,9.3)
- Regularization
  - not much discussed (see more in lecture 9.3 and e.g. https://betanalpha.github.io/assets/case_studies/bayes_sparse_regression.html)
- Unequal variances and correlations

Lasso and Bayesian lasso

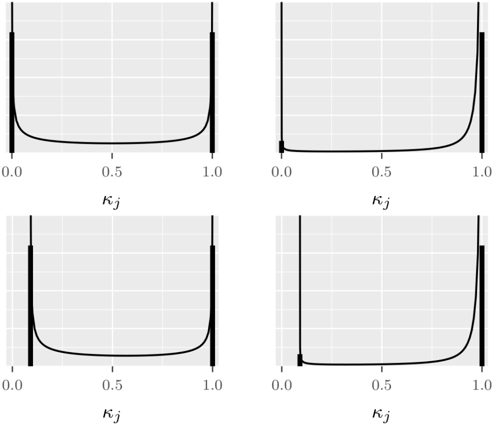
- Lasso is penalized maximum likelihood linear regression, with L1 one penalty where the amount penalty is adapted
    - penalized maximum likelihood finds the mode given the penalty parameter, and is almost the same as maximum a posteriori
    - when the amount of penalty is increased, marginal modes of weak effects go to zero first
    - when the amount of penalty is increased, also the relevant coefficients are shrunk towards zero
    - sometimes relaxed lasso is used, where after variable selection coefficients are re-estimated
- Bayesian lasso uses Laplace distribution as prior
    - Laplace prior is equivalent to L1 penalty
    - but the Bayesian inference includes distribution for parameters and that distribution doesn't shrink to a point at zero, even if the mode would be at zero
    - empirically better results obtained with more sparse priors
    - it's best to separate selection of sensible prior, good posterior inference, and the decision analysis of which variables are important

Sparse priors



from Carvalho, Polson, Scott (2009).

Regularized horseshoe



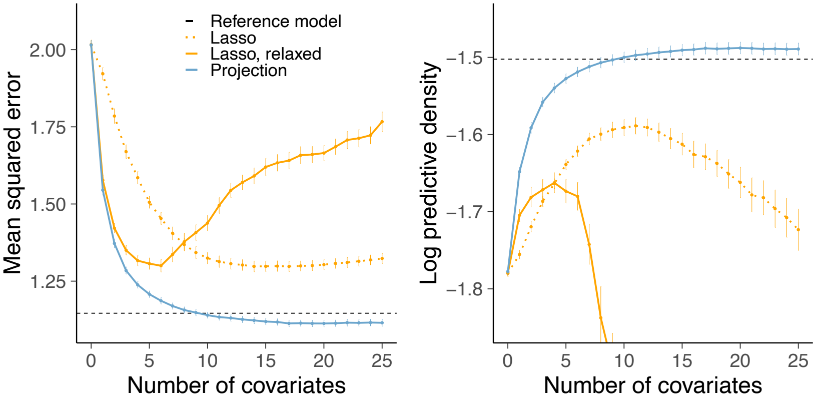
for more see

- Piironen and Vehtari (2017). Sparsity information and regularization in the horseshoe and other shrinkage priors. In Electronic Journal of Statistics, 11(2):5018-5051. Online
- https://betanalpha.github.io/assets/case_studies/bayes_sparse_regression.html

Projpred selection vs. Lasso

See projpred in lecture 9.3

Same simulated regression data as in lecture 9,3,
$n = 50$, $p = 500$, $p_{\text{rel}} = 150$, $\rho = 0.5$



Chapter 15: Hierarchical linear models

- Since you know hierarchical models, theory is easy
- With probabilistic programming computation is also easy
  - BDA3 discusses some other computational issues
  - section on transformations for HMC is relevant (see also Stan user guide 21.7 Reparameterization)
- Fixed, random, and mixed effects models
  - we don't recommend using these terms, but they are so popular that it's useful to know them

`y ~ 1 + x`	fixed / population effect; pooled model
`y ~ 1 + (0 + x | g)`	random / group effects
`y ~ 1 + x + (1 + x | g)`	mixed effects; hierarchical model

- ANOVA in section 15.6 (see also `stan_aov`)

Chapter 16: Generalized linear models

- Bioassay model is an example of GLM
- Components:
  1. The linear predictor $\eta = X\beta$
  2. The link function $g(\cdot)$ and $\mu = g^{-1}(\eta)$
  3. Outcome distribution model with location parameter μ
     - the distribution can also depend on dispersion parameter ϕ
     - originally just exponential family distributions (e.g. Poisson, binomial, negative-binomial), which all have natural location-dispersion parameterization
     - after MCMC made computation easy, GLM can refer to models where outcome distribution is not part of exponential family and dispersion parameter may have its own latent linear predictor
  - Hierarchical GLM natural extension
  - 16.3 Weakly informative priors section is excellent although the recommendation on using Cauchy has changed (see https://github.com/stan-dev/stan/wiki/Prior-Choice-Recommendations)

Chapter 17: Models for robust inference

- For example

normal	$\rightarrow$	t-distribution
Poisson	$\rightarrow$	negative-binomial
binomial	$\rightarrow$	beta-binomial
probit	$\rightarrow$	logistic / robit

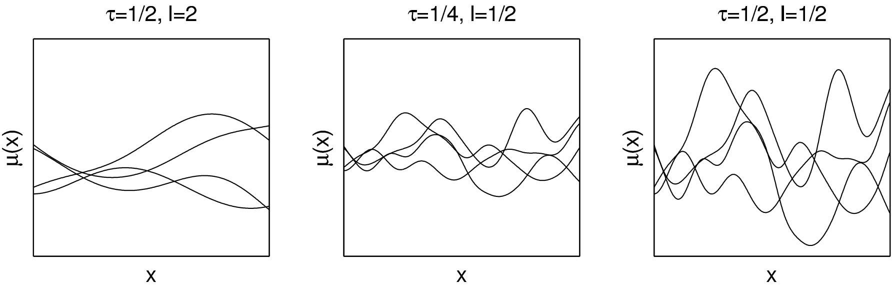
- Computation with MCMC easy
  - posterior can be multimodal
  - rstanarm doesn't have t-distribution for outcome, but brms has

Chapter 18: Models for missing data

- Extends the data collection modelling from Chapter 8
- Useful terms
  - Missing completely at random (MCAR)
    missingness does not depend on missing values or other observed values (including covariates)
  - Missing at random (MAR)
    missingness does not depend on missing values but may depend on other observed values (including covariates)
  - Missing not at random (MNAR)
    missingness depends on missing values
- Multiple imputation
  1. make a model predicting missing data
  2. sample repeatedly from the missing data model to generate multiple imputed data sets
  3. make usual inference for each imputed data set
  4. combine results

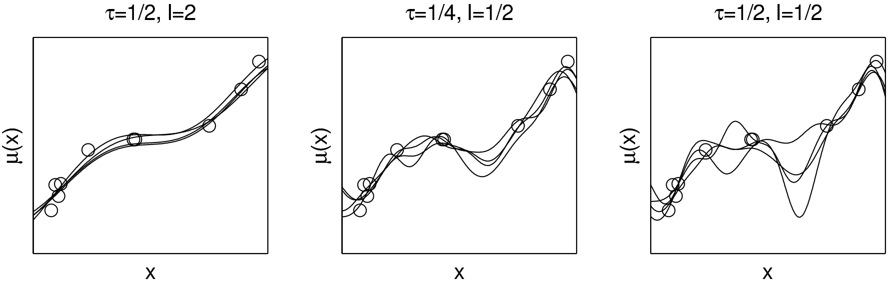
Chapter 21: Gaussian process models

- Gaussian process is
  - infinite dimensional extension of normal distribution
  - useful prior for non-linear functions
  - for any finite number of variables, the marginal is multivariate normal $f_1, \ldots, f_n \sim \mathrm{N}\left(\mu(x_1, \ldots, x_n), K(x_1, \ldots, x_n)\right)$
- Often a priori $\mu = 0$
- Prior for smooth non-linear functions, e.g. with $k(x, x') = \tau^2 \exp\left(-\frac{|x-x'|^2}{2l^2}\right)$



Chapter 21: Gaussian process models

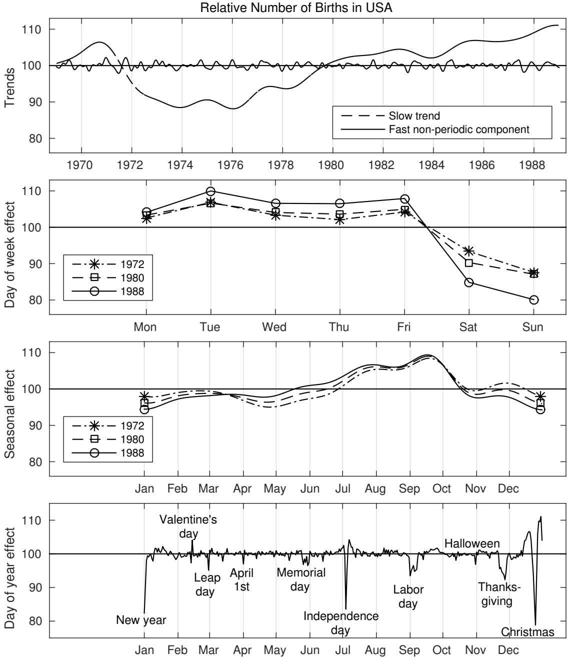
- Gaussian process is
  - infinite dimensional extension of normal distribution
  - useful prior for non-linear functions
  - for any finite number of variables, the marginal is multivariate normal $f_1, \ldots, f_n \sim \mathrm{N}\left(\mu(x_1, \ldots, x_n), K(x_1, \ldots, x_n)\right)$
- Often a priori $\mu = 0$
- Prior for smooth non-linear functions, e.g. with $k(x, x') = \tau^2 \exp\left(-\frac{|x-x'|^2}{2l^2}\right)$



Chapter 21: Gaussian process models

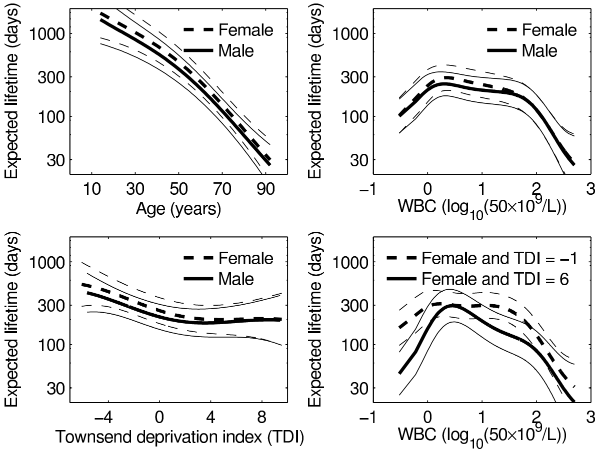
- Conditional on covariance function parameter the posterior is just multivariate normal
  - need to make inference for covariance function parameters given the marginal likelihood
  - the exact computation of the marginal likelihood scales $O(N^3)$

- Easy to make additive models
  $$y_t(t) = f_1(t) + f_2(t) + f_3(t) + f_4(t) + f_5(t) + \epsilon_t$$



Chapter 21: Gaussian process models

- For non-Gaussian outcome models similar extension as GLMs
- Survival model example:



GPs in Stan

- GP specific software (e.g. GPy, GPflow, GPyTorch) scale computationally better for GPs than Stan
- Stan has some built-in covariance functions (and soon GPU support)
- In case of non-Gaussian outcome models, sampling of latent variables can be slow (Laplace integration over the latents coming)

GPs in Stan

- GP specific software (e.g. GPy, GPflow, GPyTorch) scale computationally better for GPs than Stan
- Stan has some built-in covariance functions (and soon GPU support)
- In case of non-Gaussian outcome models, sampling of latent variables can be slow (Laplace integration over the latents coming)
- Instead of covariance matrix based approach, for low dimensional cases faster to use basis function representation
  - e.g. `stan_glm(y ~ s(x, bs="gp"))`