Bayesian Statistics for Genetics

Lecture 5: Linear models

June, 2026

In this session

- An important conjugacy result – Normal prior, Normal likelihood
- Linear regression – the standard approach for continuous outcomes (this material mostly review?)
- Bayesian linear regression
- A first look at Markov Chain Monte Carlo – much more in later sessions

Normal prior, Normal likelihood

The best-known and most widely-used statistical is the *Normal* or *Gaussian* distribution, denoted $N(\mu, \sigma^2)$ for mean μ and variance σ^2.

In sampling distributions, approximately Normal observations are expected when the variable is the sum of many similarly-sized small influences (e.g. height as a sum of SNP effects). This is called the *Central Limit Theorem*.

In priors, Normals are a convenient way to state beliefs, mainly because they're so familiar. They can also make some calculations easier (which is why Gauss liked them).

Normal prior, Normal likelihood: conjugacy

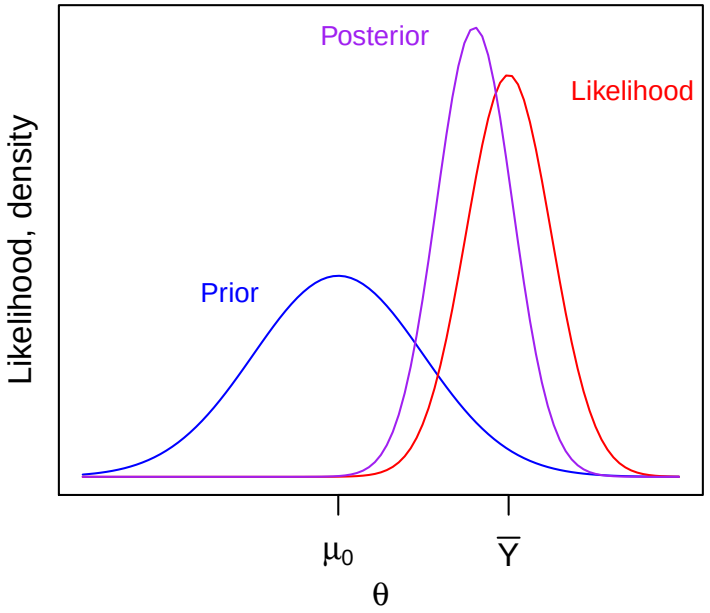
If $Y_1, \ldots Y_n$ are a random sample from $N(\theta, \sigma^2)$ with known variance σ^2:

prior $\quad \theta \sim N(\mu_0, \tau^2)$

sampling model $\quad Y \sim N(\theta, \sigma^2)$

then posterior $\quad \theta|Y \sim N\left(w\bar{Y} + (1-w)\mu_0, \frac{1}{\frac{1}{\tau^2}+\frac{n}{\sigma^2}}\right)$

where $\quad w = \frac{\frac{n}{\sigma^2}}{\frac{1}{\tau^2}+\frac{n}{\sigma^2}}$



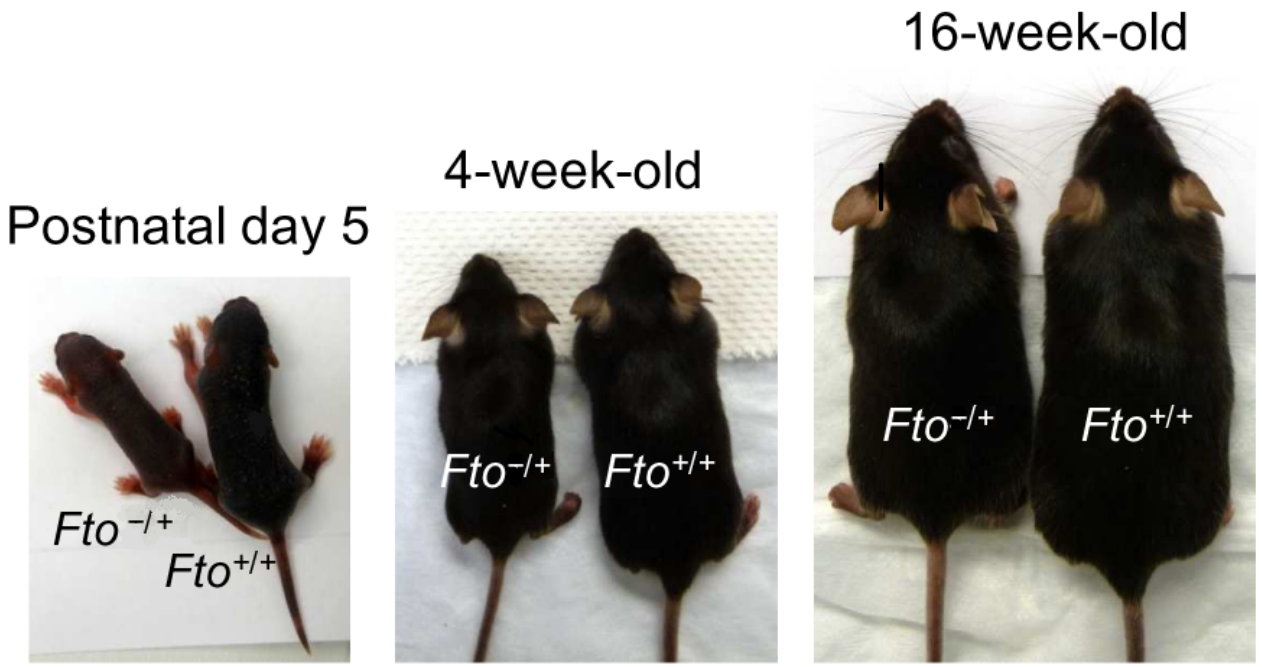
- Posterior mean is a weighted average of prior mean and sample mean
- Weights are $\propto$ precision, i.e. 1/variance.
- Posterior precision = prior precision + precision from data
- With large n Bayes gives essentially standard result: point estimate $\bar{Y}$, 95% interval is $\bar{Y} \pm 1.96\sigma/\sqrt{n}$. With tiny n, posterior$\approx$prior.

FTO example: design

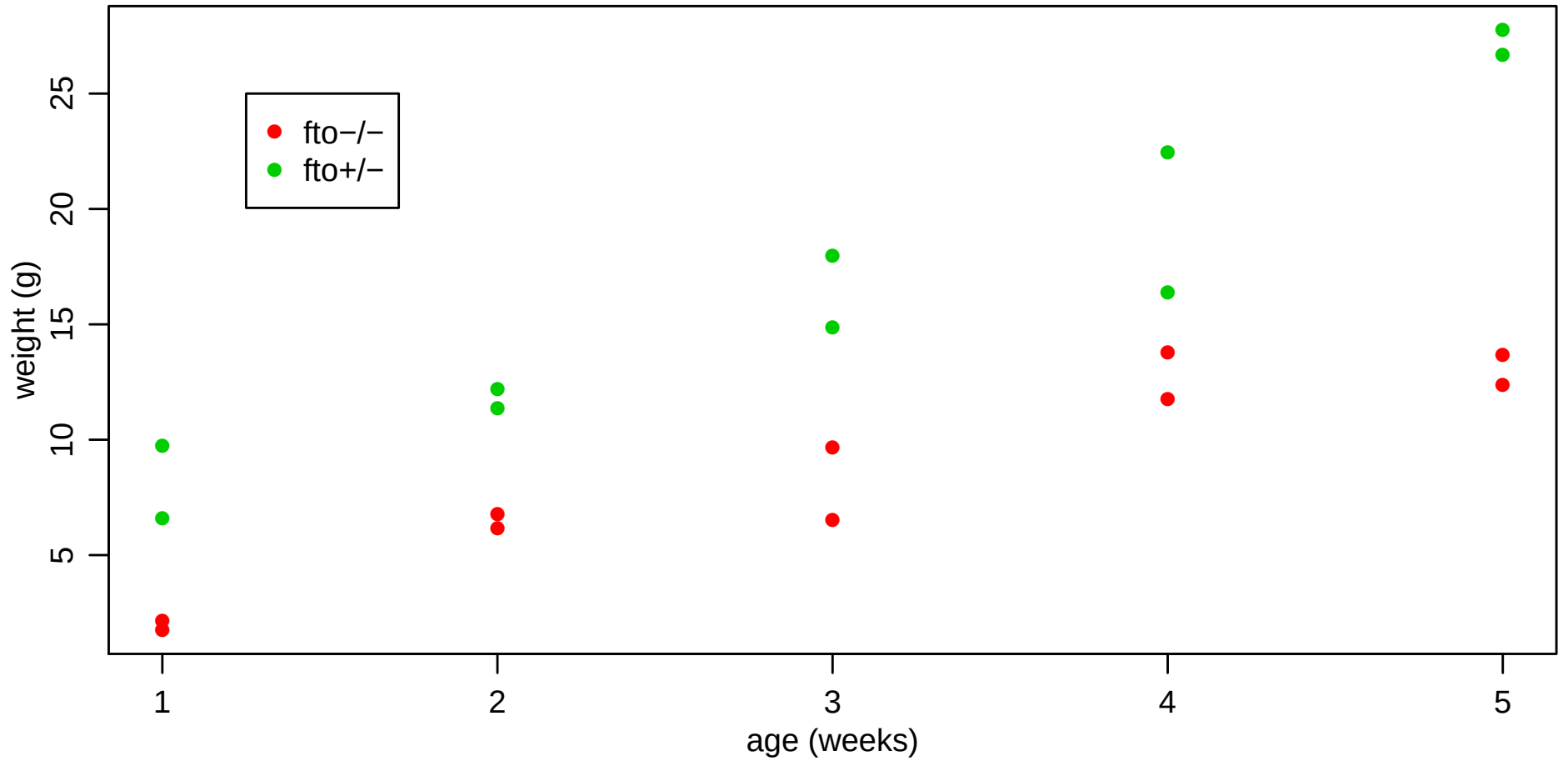
We'll soon use a version of Normal/Normal conjugacy in an analysis of the FTO gene, involved in growth and obesity.

Experimental design:

- 10 $fto+/-$ mice
- 10 $fto-/-$ mice
- Mice are sacrificed at the end of 1-5 weeks of age (not 16 weeks, as here)
- Two mice in each group are sacrificed at each age
- $n = 20$ overall, so a small but well-controlled study



FTO example: data



FTO example: critical modeling assumption

In this example:

- $Y =$ weight, which will be Normal **conditional** on covariate-specific mean
- $x_g =$ indicator of fto heterozygote $\in \{0, 1\} =$ number of "+" alleles
- $x_a =$ age in weeks $\in \{1, 2, 3, 4, 5\}$

The critical modeling choice is how mean Y depends on the covariates. Without further assumptions, we could choose to model $\mathbb{E}[\,Y|X = x\,]$ as

	age (weeks)				
genotype	1	2	3	4	5
$-/-$	$\theta_{0,1}$	$\theta_{0,2}$	$\theta_{0,3}$	$\theta_{0,4}$	$\theta_{0,5}$
$+/-$	$\theta_{1,1}$	$\theta_{1,2}$	$\theta_{1,3}$	$\theta_{1,4}$	$\theta_{1,5}$

This can't be 'wrong' – there *is* a mean weight for each genotype at each age – but we'd need to estimate 10 parameters from $n = 20$ data points.

Linear regression

But pragmatically, we instead assume smoothness in mean weight as a function of age. For each group, we assume

$$\mathbb{E}[\,Y|X_a = x_a\,] = \alpha_0 + \alpha_1 x_a$$

for some unknown α_0 and α_1, that we'll estimate.

- Now we can *borrow strength* across the ages. For example, the difference in 4-5 week mice weights tells us about the difference in 1-2 week weights
- Can also write the assumption as

  $$y = \alpha_0 + \alpha_1 x_a + \epsilon,$$

  where ϵ is some mean-zero 'noise'
- A more complex linear regression might assume e.g. $y = \alpha_0 + \alpha_1 x_a + \alpha_2 x_a^2 + \alpha_3 x_a^3 + \epsilon$, — linearity means "linear in the parameters", i.e. adding several covariates, each multiplied by its own α coefficient.

Multiple linear regression

We can also create variables, to describe models for both groups simultaneously:

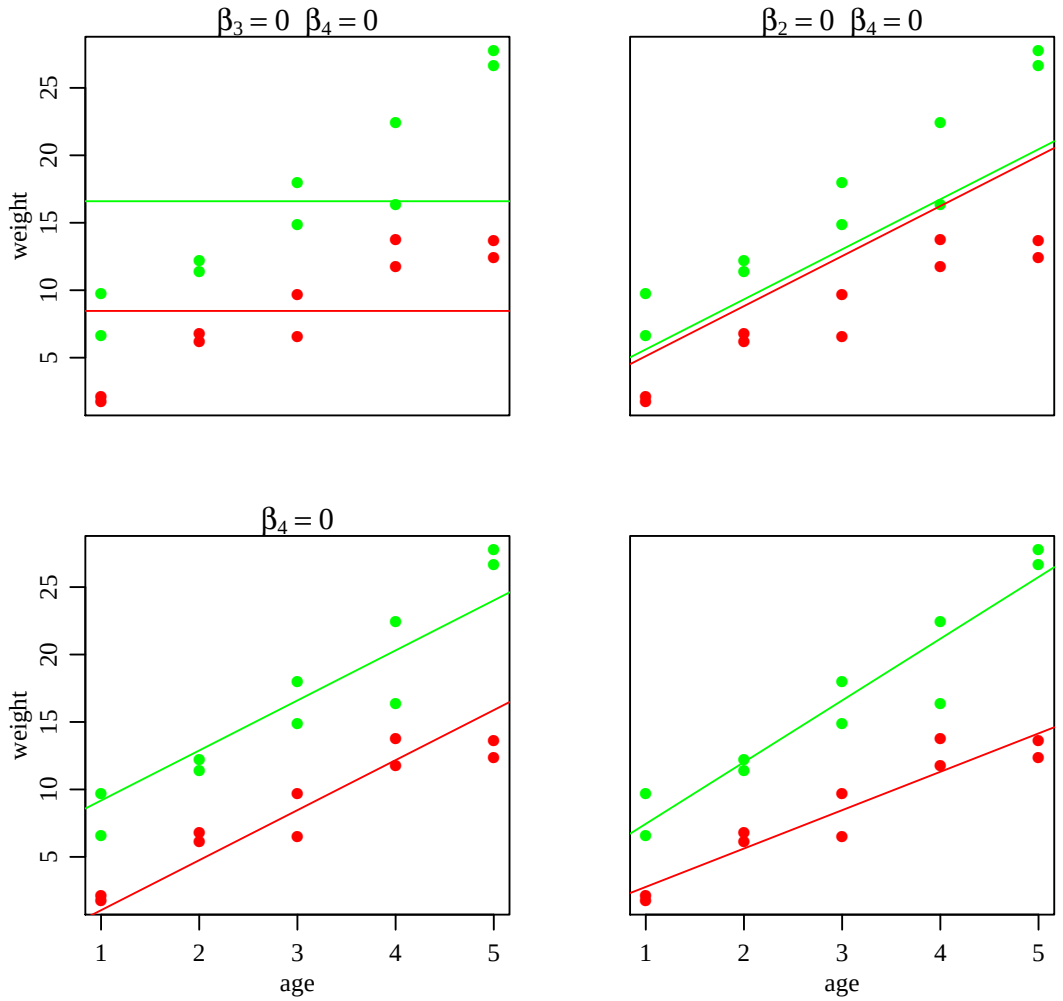
$$
\begin{aligned}
x_{i,1} &= 1 \text{ for each subject } i \\
x_{i,2} &= 0 \text{ if subject } i \text{ is homozygous, 1 if heterozygous} \\
x_{i,3} &= \text{age of subject } i, \text{ in weeks} \\
x_{i,4} &= x_{i,2} \times x_{i,3} \\
\mathbb{E}[\,Y_i|\mathbf{X} = \boldsymbol{x}\,] &= \beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_3 x_{i,3} + \beta_4 x_{i,4}, \\
\text{...or write this as } Y_i &= \beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_3 x_{i,3} + \beta_4 x_{i,4} + \epsilon_i
\end{aligned}
$$

for some mean-zero ϵ_i. Note that under this model,

$$
\begin{aligned}
\mathbb{E}[\,Y|\boldsymbol{x},] &= \beta_1 + \beta_3 \times \text{age} \quad \text{if } x_2 = 0, \text{ and} \\
\mathbb{E}[\,Y|\boldsymbol{x}\,] &= (\beta_1 + \beta_2) + (\beta_3 + \beta_4) \times \text{age} \quad \text{if } x_2 = 1.
\end{aligned}
$$

Multiple linear regression

For graphical thinkers...



Normal linear regression

How do the Y_i vary around $\mathbb{E}[\,Y_i|\boldsymbol{\beta}, \boldsymbol{x}_i\,]$, ?

$$\begin{aligned} Y_i &= \boldsymbol{\beta}^T \boldsymbol{x}_i + \epsilon_i \\ \epsilon_1, \ldots, \epsilon_n &\sim \text{i.i.d. Normal}(0, \sigma^2). \end{aligned}$$

Assuming independent identically-distributed Normal errors gives the likelihood:

$$\begin{aligned} p(y_1, \ldots, y_n | \boldsymbol{x}_1, \ldots \boldsymbol{x}_n, \boldsymbol{\beta}, \sigma^2) &= \prod_{i=1}^{n} p(y_i | \boldsymbol{x}_i, \boldsymbol{\beta}, \sigma^2) \\ &= (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T \boldsymbol{x}_i)^2 \right). \end{aligned}$$

Note: in large(r) sample sizes, what we learn about the β_j is "robust" to the Normality assumption—but with $n = 20$ we do **rely** on the mean being linear in the $\boldsymbol{x}$'s, and on the ϵ_i's variance σ^2 being constant with respect to $\boldsymbol{x}$.

Linear regression: non-Bayes methods (*)

What values of $\boldsymbol{\beta}$ are consistent with our data $\boldsymbol{y}, \mathbf{X}$? Recall that

$$p(y_1, \ldots, y_n | \boldsymbol{x}_1, \ldots \boldsymbol{x}_n, \boldsymbol{\beta}, \sigma^2) = (2\pi\sigma^2)^{-n/2} \exp\{-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T \boldsymbol{x}_i)^2\}.$$

This is big when $\text{SSR}(\boldsymbol{\beta}) = \sum (y_i - \boldsymbol{\beta}^T \boldsymbol{x}_i)^2$ is small. It can also be written as

$$\text{SSR}(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T \boldsymbol{x}_i)^2 = (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^T (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) = \boldsymbol{y}^T \boldsymbol{y} - 2\boldsymbol{\beta}^T \mathbf{X}^T \boldsymbol{y} + \boldsymbol{\beta}^T \mathbf{X}^T \mathbf{X} \boldsymbol{\beta},$$

where we use matrix notation, i.e.

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \mathbf{X}\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{x}_1 \rightarrow \\ \boldsymbol{x}_2 \rightarrow \\ \vdots \\ \boldsymbol{x}_n \rightarrow \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^T \boldsymbol{\beta} \\ \boldsymbol{x}_2^T \boldsymbol{\beta} \\ \vdots \\ \boldsymbol{x}_n^T \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \beta_1 x_{1,1} + \cdots + \beta_p x_{1,p} \\ \beta_2 x_{2,1} + \cdots + \beta_p x_{2,p} \\ \vdots \\ \beta_1 x_{n,1} + \cdots + \beta_p x_{n,p} \end{pmatrix}.$$

Linear regression: non-Bayes methods (*, almost)

Classical linear regression uses the $\hat{\boldsymbol{\beta}}$ that makes SSR smallest, i.e. finds the 'best-fitting' model. To minimize SSR, 'recall' that...

1. a minimum of a function $g(z)$ occurs at a value z such that $\frac{d}{dz}g(z) = 0$;
2. the derivative of $g(z) = az$ is a and the derivative of $g(z) = bz^2$ is $2bz$.

Doing this for SSR...

$$\begin{aligned}
\frac{d}{d\boldsymbol{\beta}}\text{SSR}(\boldsymbol{\beta}) &= \frac{d}{d\boldsymbol{\beta}}\left(\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{\beta}^T\mathbf{X}^T\boldsymbol{y} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}\right) \\
&= -2\mathbf{X}^T\boldsymbol{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}\ , \\
\text{and so } \frac{d}{d\boldsymbol{\beta}}\text{SSR}(\boldsymbol{\beta}) = 0 &\Leftrightarrow -2\mathbf{X}^T\boldsymbol{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = 0 \\
&\Leftrightarrow \mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\boldsymbol{y} \Leftrightarrow \boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}\,.
\end{aligned}$$

$\hat{\boldsymbol{\beta}}_{\text{ols}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}$ is the *Ordinary Least Squares* (OLS) estimator of $\boldsymbol{\beta}$.

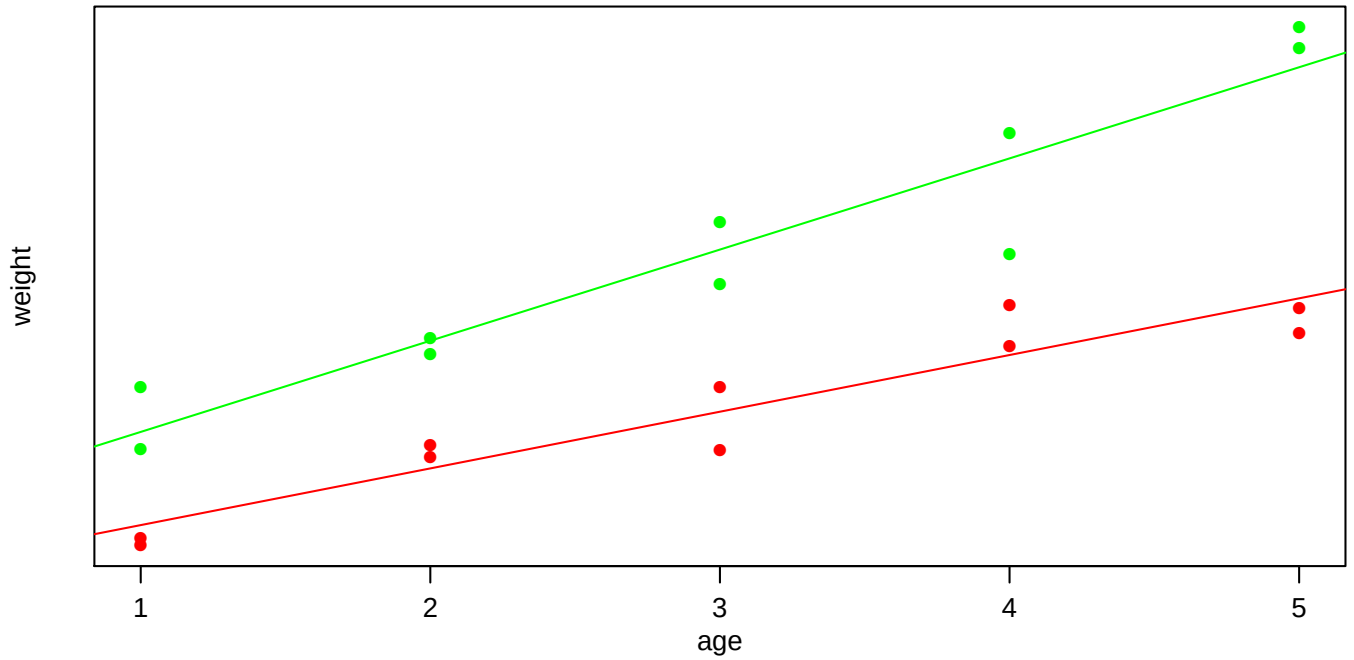
Linear regression: non-Bayes methods in R

```
### OLS estimate
> beta.ols <- solve( t(X)%*%X )%*%t(X)%*%y
> beta.ols
                    [,1]
(Intercept) -0.06821632
xg           2.94485495
xa           2.84420729
xg:xa        1.72947648

### or less painfully, using lm() and R's formula syntax:
> fit.ols<-lm( y~ xg*xa )

> coef( summary(fit.ols) )
               Estimate Std. Error     t value     Pr(>|t|)
(Intercept) -0.06821632  1.4222970 -0.04796208 9.623401e-01
xg           2.94485495  2.0114316  1.46405917 1.625482e-01
xa           2.84420729  0.4288387  6.63234803 5.760923e-06
xg:xa        1.72947648  0.6064695  2.85171239 1.154001e-02
```

Linear regression: non-Bayes methods in R



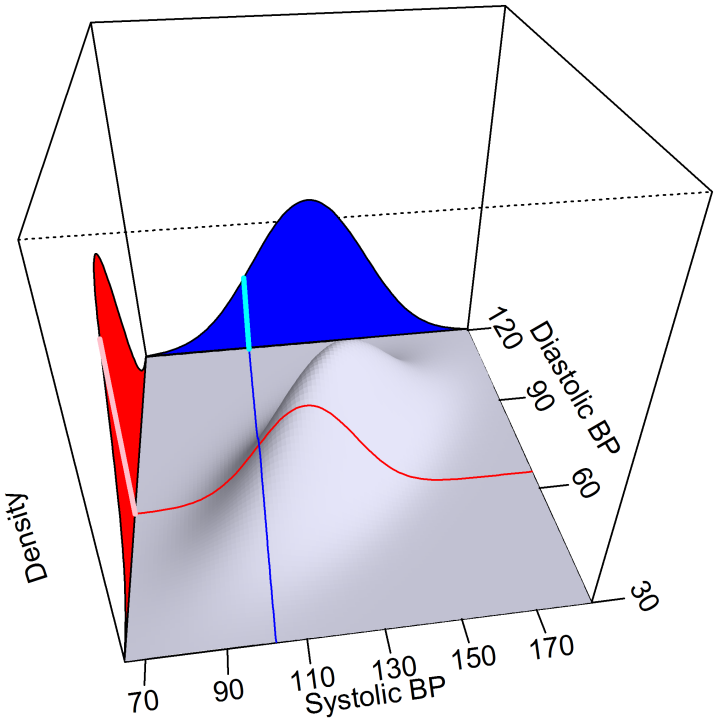

```
> coef( summary(fit.ols) )
               Estimate Std. Error     t value     Pr(>|t|)
(Intercept) -0.06821632  1.4222970 -0.04796208 9.623401e-01
xg           2.94485495  2.0114316  1.46405917 1.625482e-01
xa           2.84420729  0.4288387  6.63234803 5.760923e-06
xg:xa        1.72947648  0.6064695  2.85171239 1.154001e-02
```

(The classic version estimates the standard error of each $\widehat{\beta}_j$, and gives p-values indicating how significantly they differ from zero. t_j denotes $\widehat{\beta}_j$/Est.StdErr[$\widehat{\beta}_j$], which has a t distribution if β_j=0.)

Multivariate Normal distribution

Before Bayesian linear regression, a final piece of math notation.

The *multivariate Normal* distribution describes random vectors. For mean $\boldsymbol{m}$ and variance matrix* $\mathbf{V}$, it is written mvn$(\boldsymbol{m}, \mathbf{V})$ – see right for an example:



So we can write a full linear regression model as just

$$\boldsymbol{y}|\mathbf{X}, \boldsymbol{\beta}, \sigma^2 \sim \text{mvn}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}),$$

where σ^2 is the (assumed-constant) variance of each observation around its mean, and $\boldsymbol{I}$ denotes an *identity* matrix, with 1s on the diagonal and 0 elsewhere.

* For random vector Y_i, variance matrix Var$[\mathbf{Y}]$ has diagonal elements Var$[Y_i]$ and off diagonal elements Cov$[Y_i, Y_j]$

Bayesian inference for linear regression models

With σ^2 known, Normal priors for $\boldsymbol{\beta}$ are conjugate. With

$$
\begin{array}{lrcl}
\text{prior} & \boldsymbol{\beta} & \sim & \text{mvn}(\boldsymbol{\beta}_0, \Sigma_0) \\
\text{sampling model} & \boldsymbol{y} & \sim & \text{mvn}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}) \\
 & & & \\
\text{then posterior} & \boldsymbol{\beta}|\boldsymbol{y}, \mathbf{X} & \sim & \text{mvn}(\boldsymbol{\beta}_n, \Sigma_n), \\
\text{where} & \Sigma_n = \text{Var}\boldsymbol{\beta}|\boldsymbol{y}, \mathbf{X}, \sigma^2 & = & (\Sigma_0^{-1} + \mathbf{X}^T\mathbf{X}/\sigma^2)^{-1} \\
 & \boldsymbol{\beta}_n = \mathbb{E}\boldsymbol{\beta}|\boldsymbol{y}, \mathbf{X}, \sigma^2 & = & (\Sigma_0^{-1} + \mathbf{X}^T\mathbf{X}/\sigma^2)^{-1}(\Sigma_0^{-1}\boldsymbol{\beta}_0 + \mathbf{X}^T\boldsymbol{y}/\sigma^2).
\end{array}
$$

- Posterior precision (inverse-variance) is the sum of precision from prior and sampling model
- Posterior mean is a kind of weighted average of prior mean on $\widehat{\beta}_{OLS}$, with precision/inverse-variance weights
- If $\Sigma_0^{-1} \ll \mathbf{X}^T\mathbf{X}/\sigma^2$, then $\boldsymbol{\beta}_n \approx \widehat{\boldsymbol{\beta}}_{\text{ols}}$, if $\Sigma_0^{-1} \gg \mathbf{X}^T\mathbf{X}/\sigma^2$, then $\boldsymbol{\beta}_n \approx \boldsymbol{\beta}_0$ – sensibly

The g-prior

But how to pick β_0, Σ_0? A classical suggestion (Zellner, 1986) uses the *g-prior:*

$$\boldsymbol{\beta} \sim \text{mvn}(\mathbf{0}, g\sigma^2(\mathbf{X}^T\mathbf{X})^{-1}).$$

Why this variance? The variance of the OLS estimate $\hat{\boldsymbol{\beta}}_{\text{ols}}$ is

$$\text{Var}\hat{\boldsymbol{\beta}}_{\text{ols}} = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} = \frac{\sigma^2}{n}(\mathbf{X}^T\mathbf{X}/n)^{-1},$$

which can be thought of as (roughly) the uncertainty in $\boldsymbol{\beta}$ from n observations. In the g-prior the variance is

$$g\sigma^2(\mathbf{X}^T\mathbf{X})^{-1} = \frac{\sigma^2}{n/g}(\mathbf{X}^T\mathbf{X}/n)^{-1},$$

which can be viewed as the uncertainty from n/g observations.

For example, $g = n$ means the prior has the same amount of info as 1 observation – so (roughly!) not much, in a large study.

Posterior distributions under the g-prior

Following the conjugacy results, with σ^2 known it turns out that

$$\boldsymbol{\beta}|\boldsymbol{y}, \mathbf{X}, \sigma^2 \sim \text{mvn}(\boldsymbol{\beta}_n, \Sigma_n),$$

$$\begin{aligned} \text{where } \Sigma_n = \text{Var}[\boldsymbol{\beta}|\boldsymbol{y}, \mathbf{X}, \sigma^2] &= \frac{g}{g+1}\sigma^2(\mathbf{X}^T\mathbf{X})^{-1} \\ \boldsymbol{\beta}_n = \mathbb{E}[\boldsymbol{\beta}|\boldsymbol{y}, \mathbf{X}, \sigma^2] &= \frac{g}{g+1}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y} \end{aligned}$$

- The posterior mean estimate $\boldsymbol{\beta}_n$ is simply $\frac{g}{g+1}\hat{\boldsymbol{\beta}}_{\text{ols}}$.
- The posterior variance of $\boldsymbol{\beta}$ is simply $\frac{g}{g+1}\text{Var}\hat{\boldsymbol{\beta}}_{\text{ols}}$.
- $\boldsymbol{\beta}_n^T\Sigma_n^{-1}\boldsymbol{\beta}_n$ is $\frac{g}{g+1}$ times smaller than $\hat{\boldsymbol{\beta}}_{OLS}^T\text{Var}[\hat{\boldsymbol{\beta}}_{OLS}]^{-1}\hat{\boldsymbol{\beta}}_{OLS}$, the usual signal-to-noise measure we turn into p-values
- g shrinks the coefficients towards $\mathbf{0}$ and so can prevent overfitting to the data
- If $g \propto n$, then as n increases, inference approximates classical methods.

Monte Carlo simulation

But σ^2 is rarely known exactly, so more realistic analysis captures that uncertainty via a prior. A convenient & popular choice is the *inverse-Gamma* distribution:

$$\begin{array}{lrcl} \text{prior} & 1/\sigma^2 & \sim & \text{gamma}(\nu_0/2, \nu_0\sigma_0^2/2) \\ \text{sampling model} & \boldsymbol{y} & \sim & \text{mvn}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}) \\ \text{posterior} & 1/\sigma^2|\boldsymbol{y},\mathbf{X} & \sim & \text{gamma}([\nu_0+n]/2, [\nu_0\sigma_0^2 + \text{SSR}_g]/2) \end{array}$$

...where SSR_g is complicated but basically a sum of squared residuals.

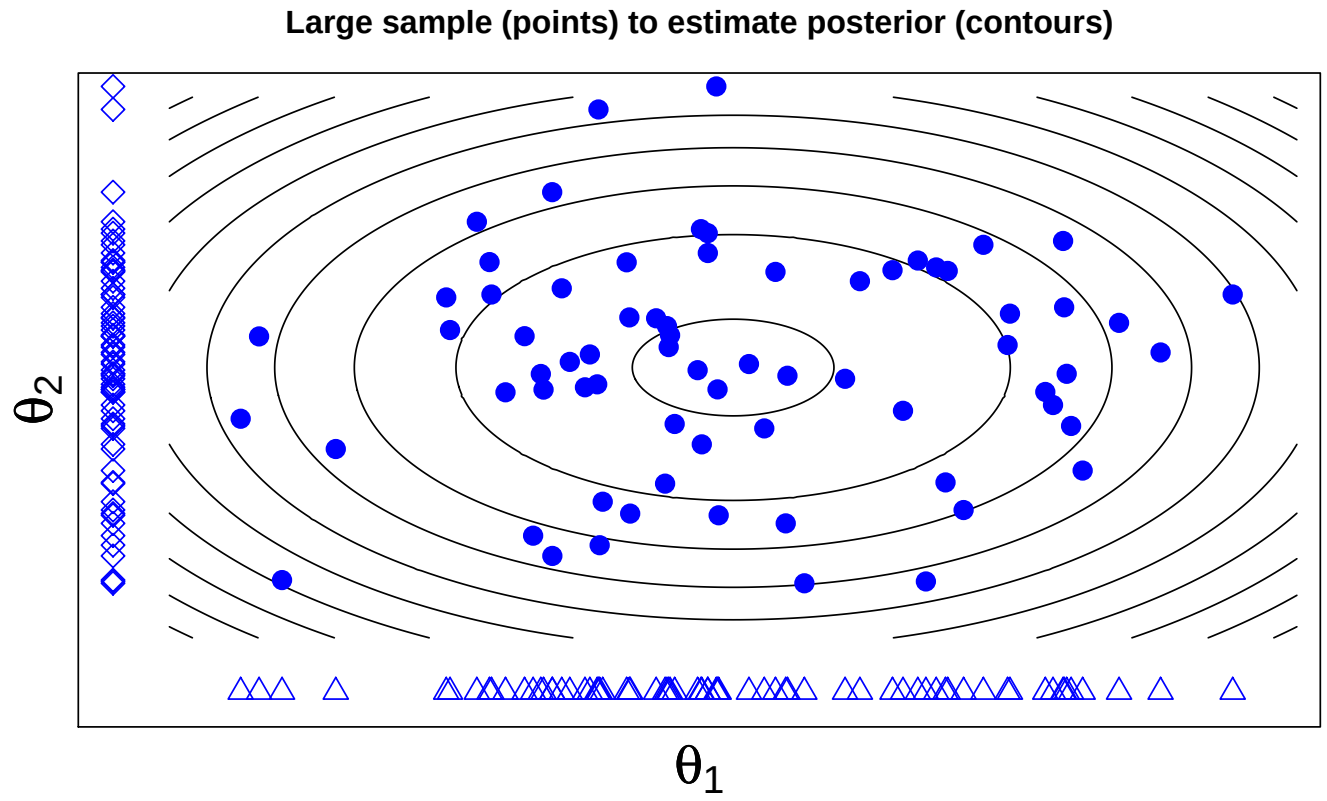
A natural next step is be to use the conjugate posteriors for σ and $\boldsymbol{\beta}$ together – usually to get the marginal posterior for $\boldsymbol{\beta}$. But this requires integration:

$$p(\boldsymbol{\beta}|\boldsymbol{y},\mathbf{X}) = \int_{\sigma^2>0} p(\boldsymbol{\beta},\sigma^2|\boldsymbol{y},\mathbf{X})d\sigma^2 = \int_{\sigma^2>0} p(\boldsymbol{\beta}|\sigma^2,\boldsymbol{y},\mathbf{X})p(\sigma^2|\boldsymbol{y},\mathbf{X})d\sigma^2$$

This takes work, and is error-prone. Instead we'll use Monte Carlo methods.

Monte Carlo simulation

The *Monte Carlo method* is important for Bayesian work;

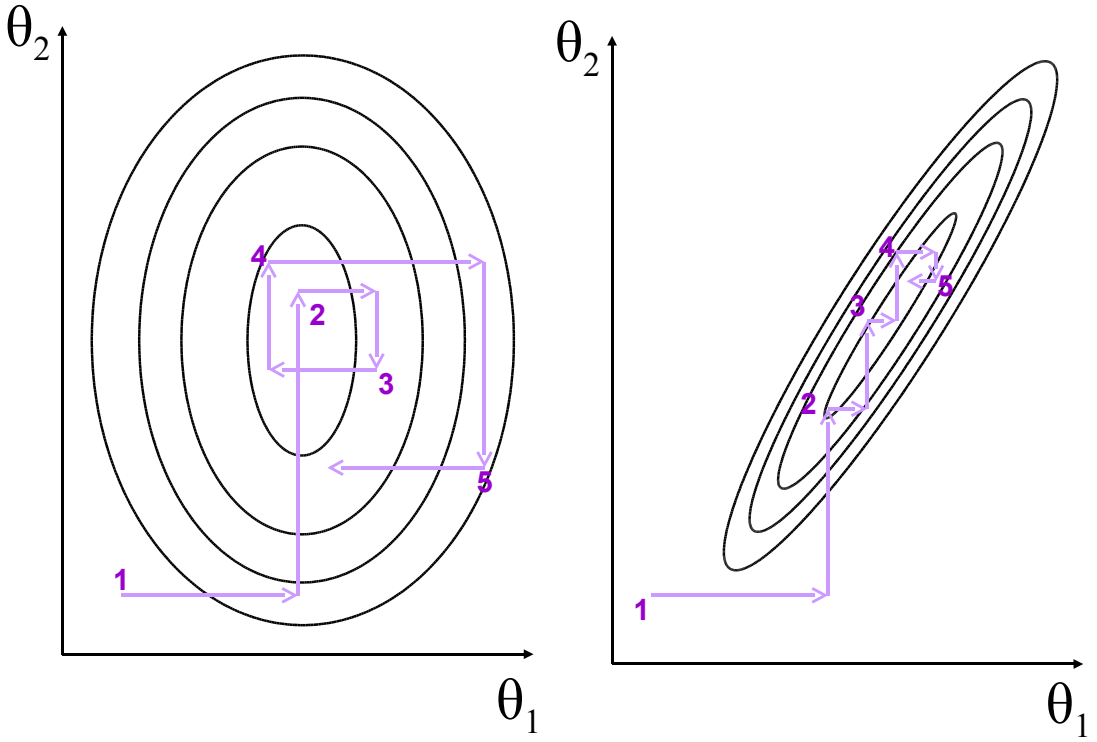


The big idea: with a large sample from a distribution – e.g. the posterior – we can approximate **any** property of that distribution.

Monte Carlo simulation

It may *seem* like sampling would require full knowledge of the posterior, and be at least as much work as the integration.

But this isn't true: one way to do it is the *Gibbs Sampler* – here shown for two examples



Gibbs updates parameters 'one at a time', using only $p(\theta_1|\theta_2)$, then $p(\theta_2|\theta_1)$ – so no hard integration to do. The sequence of samples $\boldsymbol{\theta}^{(s)}$ (a *Markov Chain*) are autocorrelated, i.e. dependent, but the posterior **is** covered appropriately.

Monte Carlo simulation

Recall the posterior can be written as $p(\theta_1, \theta_2|\mathbf{Y}) = p(\theta_2|\mathbf{Y}) \times p(\theta_1|\theta_2, \mathbf{Y})$, so if we can generate from conditional $p(\theta_1|\theta_2, \mathbf{Y})$ – in one variable – with frequency $p(\theta_2|\mathbf{Y})$ for each θ_2, we get a sample from the full posterior.

Gibbs does this, using *just* the conditionals, iterating between these steps:

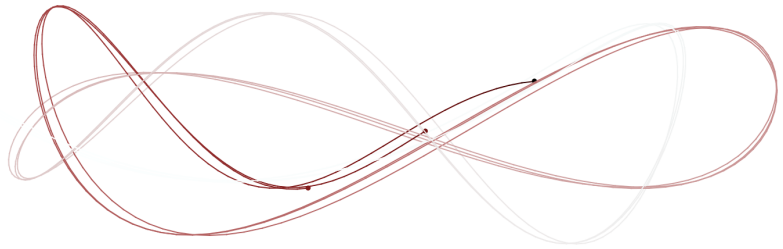
$$\begin{aligned} \theta_1^{(s)} &\sim p(\theta_1|\theta_2^{(s-1)}, \mathbf{Y}) \\ \theta_2^{(s)} &\sim p(\theta_2|\theta_1^{(s)}, \mathbf{Y}) \end{aligned}$$

to produce a *chain* $(\theta_1^{(0)}, \theta_2^{(0)}), (\theta_1^{(1)}, \theta_2^{(1)}), ..., (\theta_1^{(s)}, \theta_2^{(s)}), ...$

- If the chain is long enough ($s \to \infty$), this sequence *is* a sample from $p(\theta_1, \theta_2|\mathbf{Y})$, no matter where you started
- For more parameters, update each single θ_k in turn, then start again

MCMC: with Stan

Stan works out all the conditionals for us, and how to update based on them ;



- Specify just a model, including priors, and tell Stan what the data are
- Stan writes code to sample from the posterior, by 'walking around' – actually it runs the *No U-Turn Sampler*, a more advanced algorithm for exploring the parameter space
- Stan runs this code, and reports back all the samples
- The `rstan` package lets you run chains from R
- Some modeling limitations – no discrete parameters – but becoming very popular; works well with some models where other software would struggle
- Requires declarations (like C++) – unlike R – so models require a bit more typing...

MCMC: with Stan

For our linear regression, with unknown σ^2;

```
cat(file="FTOexample.stan", "
data {
  int n; //the number of observations
  int p; //the number of columns in the model matrix
  real y[n]; //the response
  matrix[n,p] X; //the model matrix
  real g; // Zellner scale factor
  vector[p] mu; // Zellner prior mean (all zeros)
  matrix[p,p] XtXinv; // information matrix
}
parameters {
  vector[p] beta; //the regression parameters
  real invsigma2; //the precision, a.k.a. inverse-variance
}
transformed parameters {
  vector[n] linpred;
  cov_matrix[p] Sigma;
```

MCMC: with Stan

```
  real sigma;
  linpred = X*beta;
  sigma = 1/sqrt(invsigma2);
  for (j in 1:p){
    for (k in 1:p){
      Sigma[j,k] = g*sigma^2*XtXinv[j,k];
      } } }
model {
  beta ~ multi_normal(mu, Sigma);
  y ~ normal(linpred, sigma);
  invsigma2 ~ gamma(0.5, 1.839); // we took nu0=1, sigma0=1.91
}
")
# do the MCMC, store the results
library("rstan")
stan2 <- stan(file = "FTOexample.stan",
data = list(n=n,p=p, y=y, X=X, g=n, mu=rep(0,p), XtXinv=solve(crossprod(X)) ),
iter = 100000, chains = 1, pars=c("beta","sigma"))
```

MCMC: with Stan

Summarize the posterior by summarizing the MCMC samples:

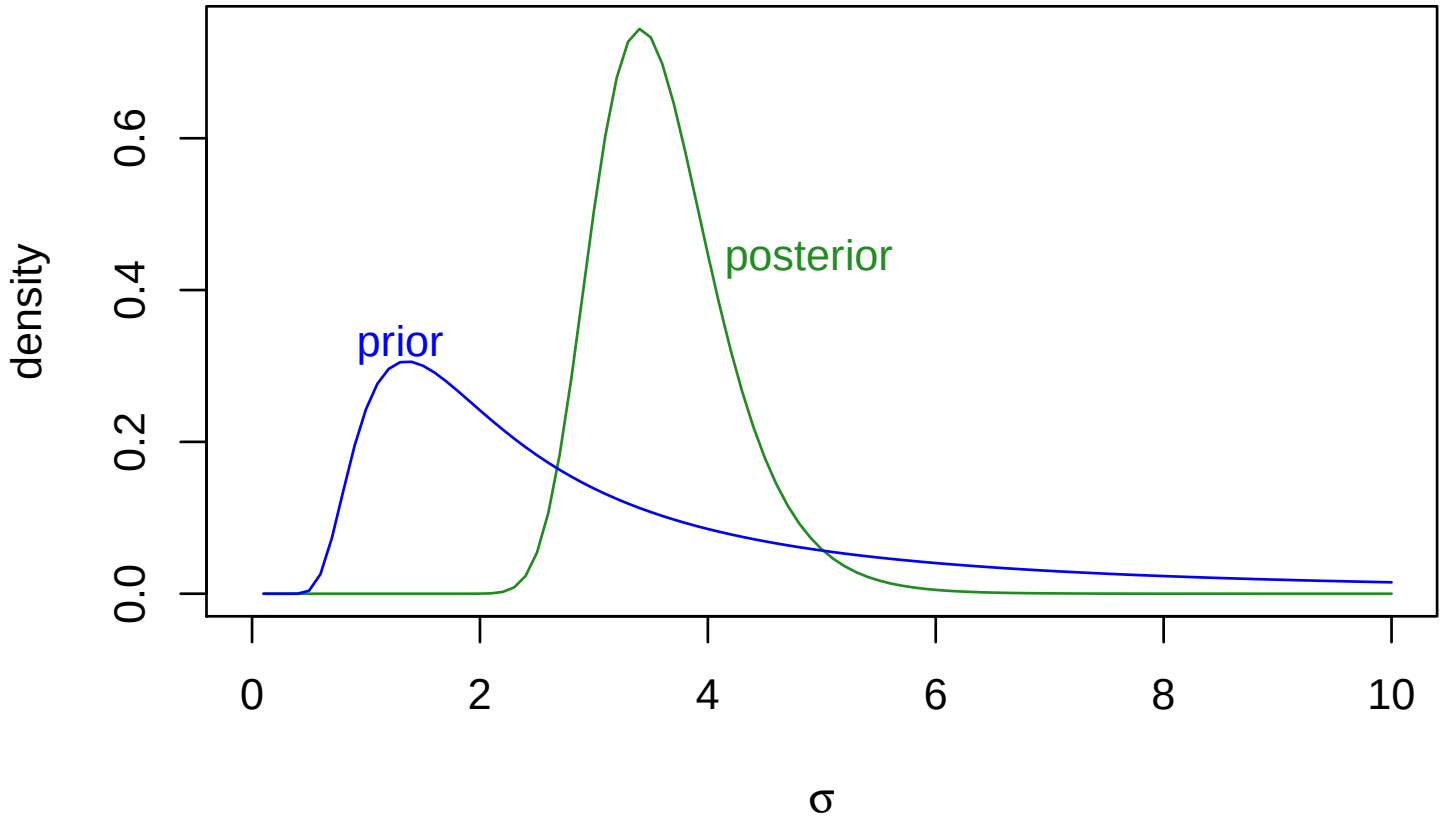
```
> print(stan2)
Inference for Stan model: FTOexample.
1 chains, each with iter=1e+05; warmup=50000; thin=1;
post-warmup draws per chain=50000, total post-warmup draws=50000.
          mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
beta[1]   0.50    0.02 2.47  -4.39  -1.11   0.50   2.10   5.39 13849    1
beta[2]   1.48    0.03 3.50  -5.42  -0.80   1.47   3.78   8.43 13720    1
beta[3]   2.13    0.01 0.75   0.65   1.64   2.13   2.61   3.61 14000    1
beta[4]   2.56    0.01 1.06   0.46   1.87   2.56   3.24   4.65 14015    1
sigma     3.39    0.00 0.55   2.52   3.00   3.32   3.70   4.66 20814    1
lp__    -40.95    0.01 1.64 -45.04 -41.79 -40.61 -39.75 -38.79 16583    1

> # compare with the non-Bayes version
> round(cbind(summary(lm(y~xg*xa))$coef[,1:2], confint(lm(y~xg*xa))), 2)
            Estimate Std. Error 2.5 % 97.5 %
(Intercept)     0.49       1.06 -1.75   2.73
xg              1.59       1.49 -1.57   4.76
xa              2.24       0.32  1.57   2.92
xg:xa           2.68       0.45  1.72   3.63
```

MCMC: with Stan

Why are the estimates similar but the intervals so different?

Here are the prior and posterior for σ;

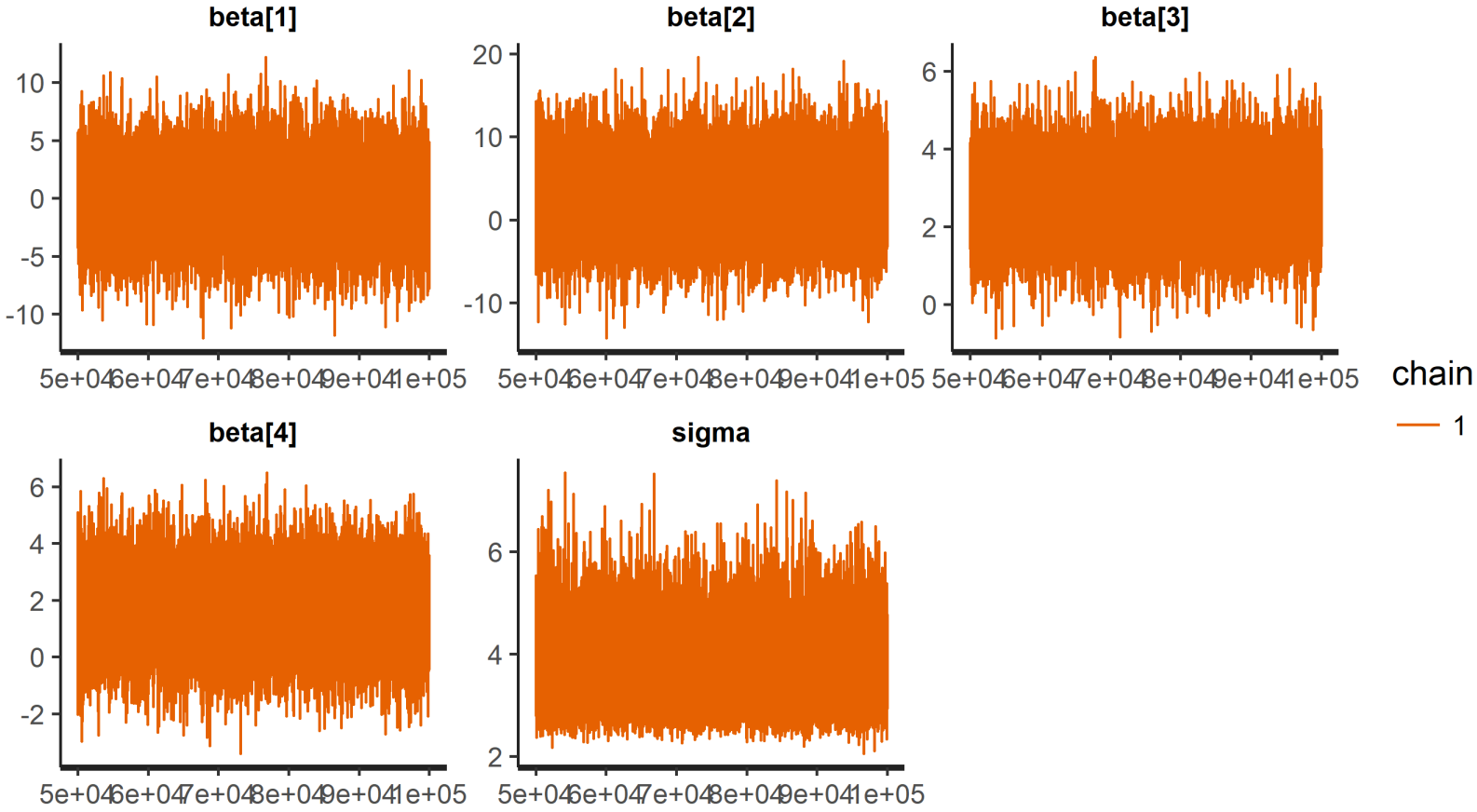


The 'best guess' estimate is $\hat{\sigma} = \sigma_0 = 1.91$ – but the prior also supports much larger values – with which the data don't strongly disagree.

MCMC: with Stan

To see where the 'chain' went...

> `traceplot(stan2)`

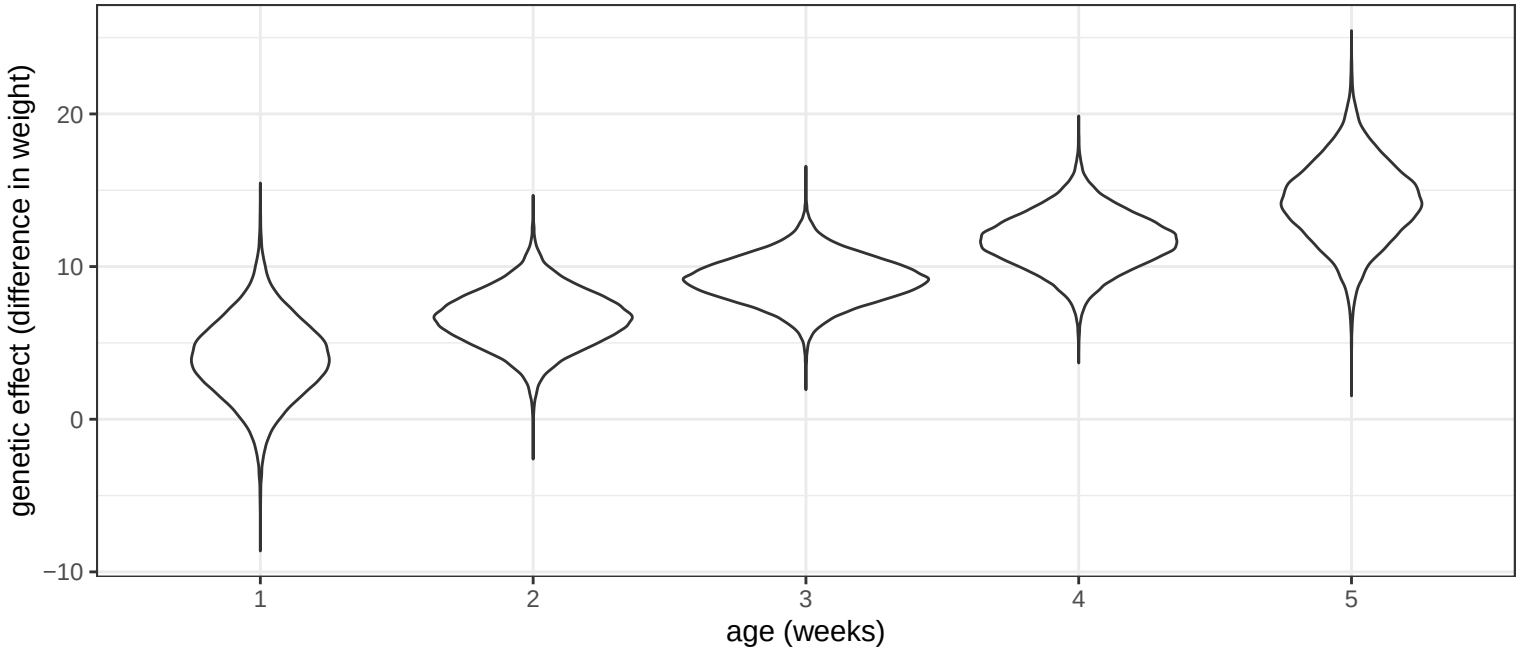


Tip: save these plots as PNGs not PDFs!

MCMC: with Stan

With the posterior samples in R, we can evaluate the posterior of any function of them. For example, the genetic effect at each of the five ages;

$$\mathbb{E}[\,Y|\text{age},+/-\,]-\mathbb{E}[\,Y|\text{age},-/-\,]=(\beta_1+\beta_2)+(\beta_3+\beta_4)\times\text{age}-(\beta_1+\beta_3\times\text{age})=\beta_2+\beta_4\times\text{age}$$



MCMC: summary so far

- Stan - and other similar software - may look like overkill for this 'conjugate' problem, but Stan can provide posteriors for almost any model
- The 'modeling' language is based on R
- Users do have to decide how long a chain to run, and how long to 'burn in' for at the start of the chain. These are not easy to answer! We'll see some diagnostics in later sessions

Summary

- Linear regressions are of great applied interest
- Conjugacy helps us quickly explore how informative the prior is, versus the data
- ...but with MCMC we can fit a huge variety of models
- Inference using MCMC output is also flexible; no restriction on parameters we can learn about

Bonus tracks: what if the model's wrong? (***)

Different types of violation—in decreasing order of how much they typically matter in practice

- Just have the wrong data (!) i.e. not the data you claim to have
- Observations are not independent, e.g. repeated measures on same mouse over time
- Mean model is incorrect
- Error terms do not have constant variance
- Error terms are not Normally distributed

Having the wrong data, where possible, should be addressed before analyses. Here we'll focus on how much post-analysis violations matter.

Wrong model: dependent outcomes

- Observations from the same mouse are more likely to be similar than those from different mice (even if they have same age and genotype)
- SBP from subjects (even with same age, genotype etc) in the same family are more likely to be similar than those in different familes – perhaps unmeasured common diet?
- Spatial and temporal relationships also tend to induce correlation

If the pattern of relationship is known, can allow for it – typically in "random effects modes" – see later session.

If not, treat results with caution! Precision is likely over-stated.

Wrong model: mean model

Even when the scientific background is highly informative about the variables of interest (e.g. we want to know about the association of Y with $\boldsymbol{x}_1$, adjusting for $\boldsymbol{x}_2$, $\boldsymbol{x}_3$...) there is rarely strong information about the form of the model

- Does mean weight increase with age? age^2? age^3?
- Could the effect of genotype also change non-linearly with age?

Including quadratic terms is a common approach – but quadratics are sensitive to the tails. Instead, including "spline" representations of covariates allows the model to capture many patterns.

Including interaction terms (as we did with $x_{i,2} \times x_{i,3}$) lets one covariate's effect vary with another.

(Deciding which covariates to use is addressed in the Model Choice session.)

Wrong model: non-constant variance

This is plausible in many situations; perhaps e.g. young mice are harder to measure, i.e. more variables. Or perhaps the FTO variant affects weight regulation — again, more variance.

- Having different variances at different covariate values is known as *heteroskedasticity*
- Unaddressed, it can result in over- or under-statement of precision

The most obvious approach is to model the variance, i.e.

$$\begin{aligned} Y_i &= \boldsymbol{\beta}^T \boldsymbol{x}_i + \epsilon_i, \\ \epsilon_i &\sim \text{Normal}(0, \sigma_i^2), \end{aligned}$$

where σ_i depends on covariates, e.g. σ_{homozy} and $\sigma_{heterozy}$ for the two genotypes.

Of course, these parameters need priors. Constraining variances to be positive also makes choosing a model difficult in practice.

Robust standard errors (in Bayes)

In linear regression, some robustness to model-misspecification and/or non-constant variance is available – but it relies on interest in linear 'trends'. Formally, we can define parameter

$$\boldsymbol{\theta} = \text{argmin } \mathbb{E}_x \left[\left(\mathbb{E}_y[y|x] - \mathbf{X}^t \boldsymbol{\theta} \right)^2 \right],$$

i.e. the straight line that best-captures random-sampling, in a least-squares sense.

- This 'trend' can capture important features in how the mean y varies at different x
- Fitting extremely flexible Bayesian models, we get a posterior for $\boldsymbol{\theta}$
- The posterior mean approaches $\widehat{\boldsymbol{\beta}}_{\text{ols}}$, in large samples
- The posterior variance approaches the 'robust' *sandwich estimate*, in large samples (details in Szpiro et al, 2011)

Robust standard errors (in Bayes)

The OLS estimator can be written as $\hat{\boldsymbol{\beta}}_{\text{ols}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y} = \sum_{i=1}^n c_i y_i$, for appropriate c_i.

$$\begin{aligned}
\textbf{True} \text{ variance} \quad \text{Var}[\,\hat{\beta}\,] &= \sum_{i=1}^n c_i^2 \text{Var}[\,Y_i\,] \\
\textbf{Robust} \text{ estimate} \quad \widehat{\text{Var}}_R[\,\hat{\beta}\,] &= \sum_{i=1}^n c_i^2 e_i^2 \\
\textbf{Model-based} \text{ estimate} \quad \widehat{\text{Var}}_M[\,\hat{\beta}\,] &= \text{Mean}(e_i^2) \sum_{i=1}^n c_i^2,
\end{aligned}$$

where $e_i = y_i - \boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}_{\text{ols}}$, the residuals from fitting a linear model.

Non-Bayesian sandwich estimates are available through R's `sandwich` package – much quicker than Bayes with a very-flexible model. For correlated outcomes, see the `GEE` package for generalizations.

Wrong model: Non-Normality

This is not a big problem for learning about population parameters;

- The variance statements/estimates we just saw don't rely on Normality
- The *central limit theorem* means that $\hat{\beta}$ ends up Normal anyway, in large samples
- In small samples, expect to have limited power to detect non-Normality
- ... except, perhaps, for extreme outliers (data errors?)

For prediction – where we assume that outcomes do follow a Normal distribution – this assumption is more important.