

LAPLACE APPROXIMATION

BAYESIAN STATISTICS FOR ECOLOGISTS

IGB 12. TO 19. NOVEMBER 2018

MCMC METHODS ARE SLOW

- ▶ Markov chains take time to learn the distribution
- ▶ With lots of parameters, memory requirements can be intense
- ▶ Sometimes, fast and approximate is good enough

LAPLACE APPROXIMATION

LAPLACE APPROXIMATION

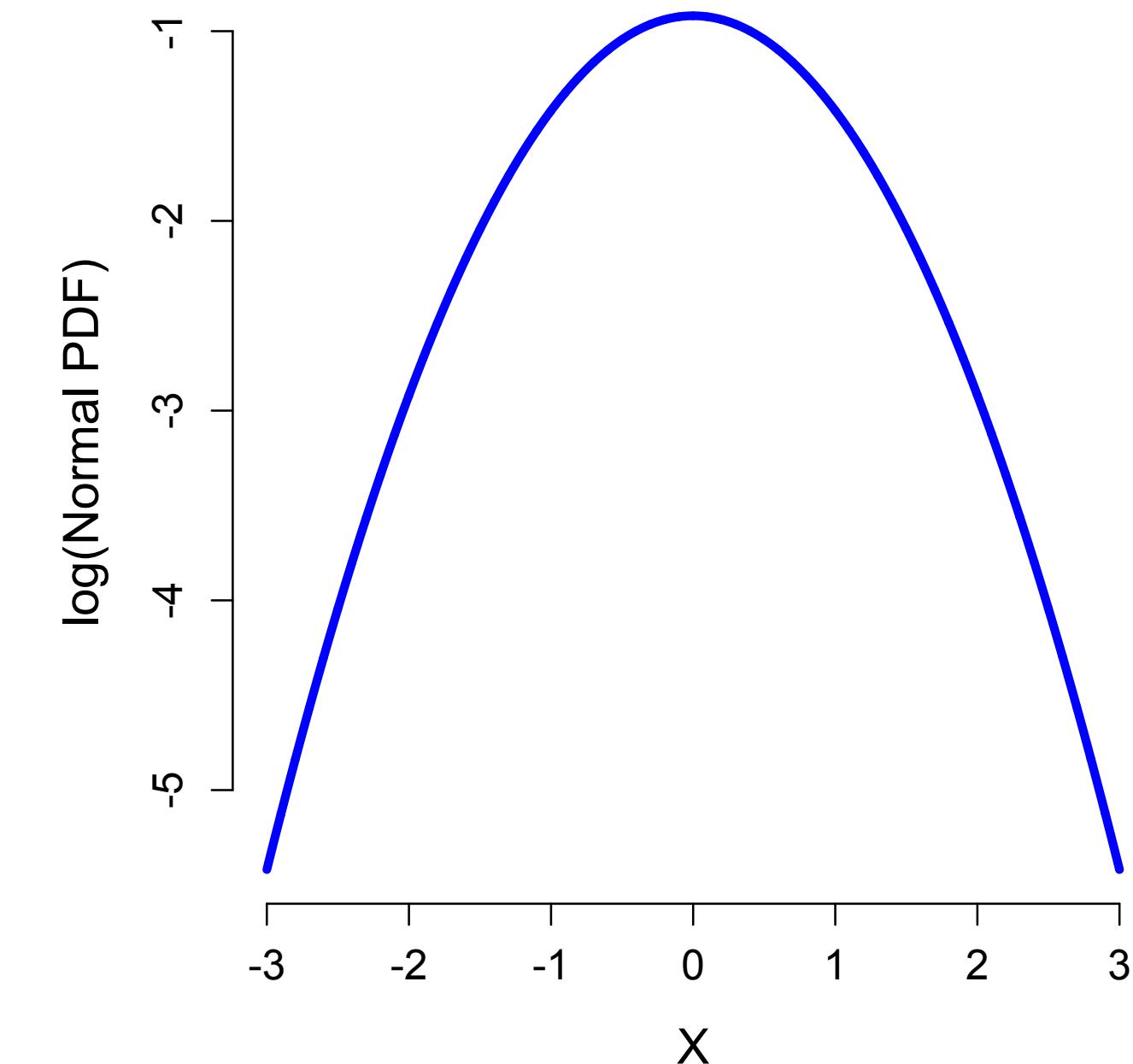
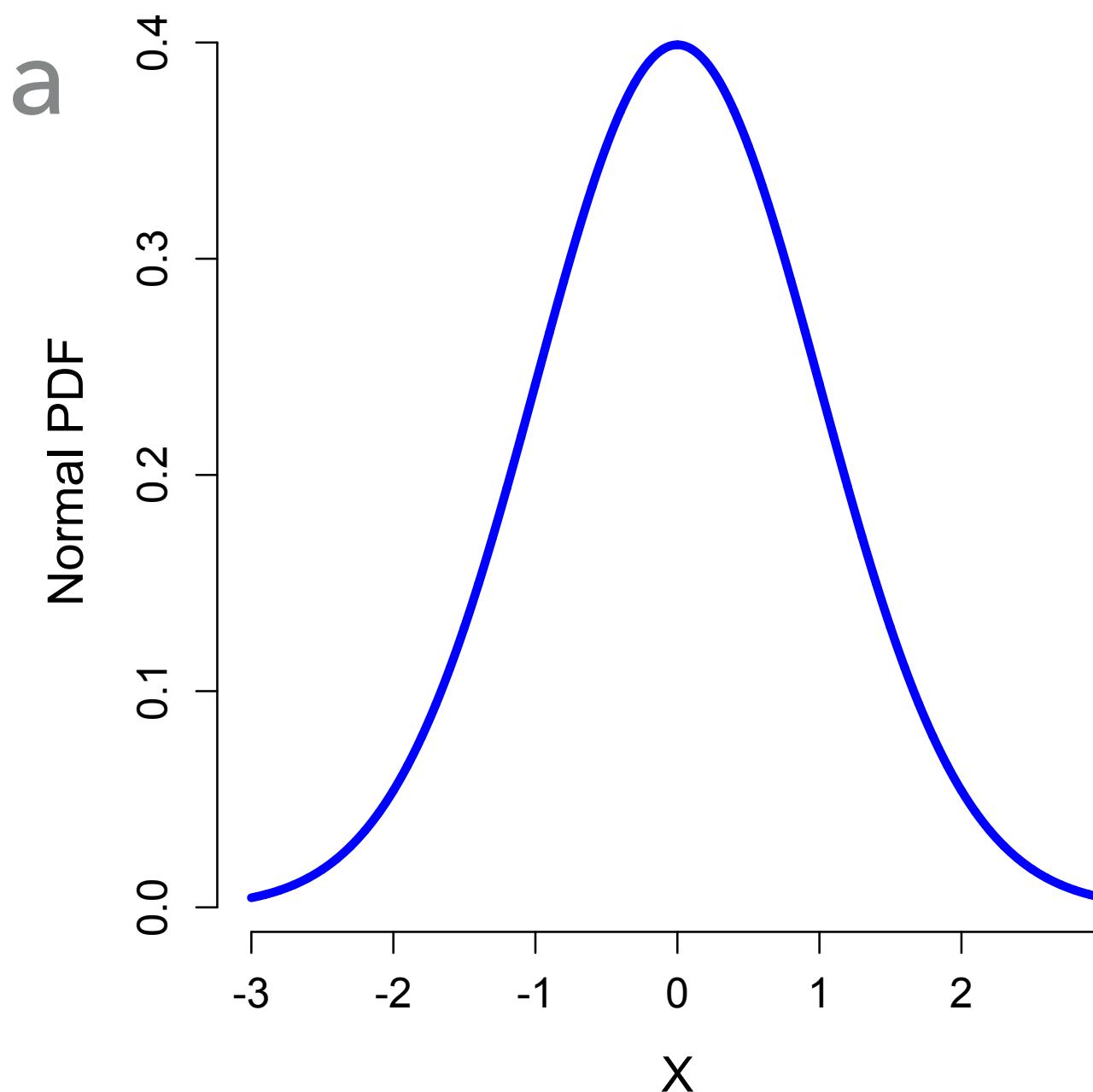
- ▶ This is a general method for approximating the shape of any curve

LAPLACE APPROXIMATION

- ▶ This is a general method for approximating the shape of any curve
- ▶ Because of the central limit theorem, many joint posteriors are approximately multivariate normal distributions

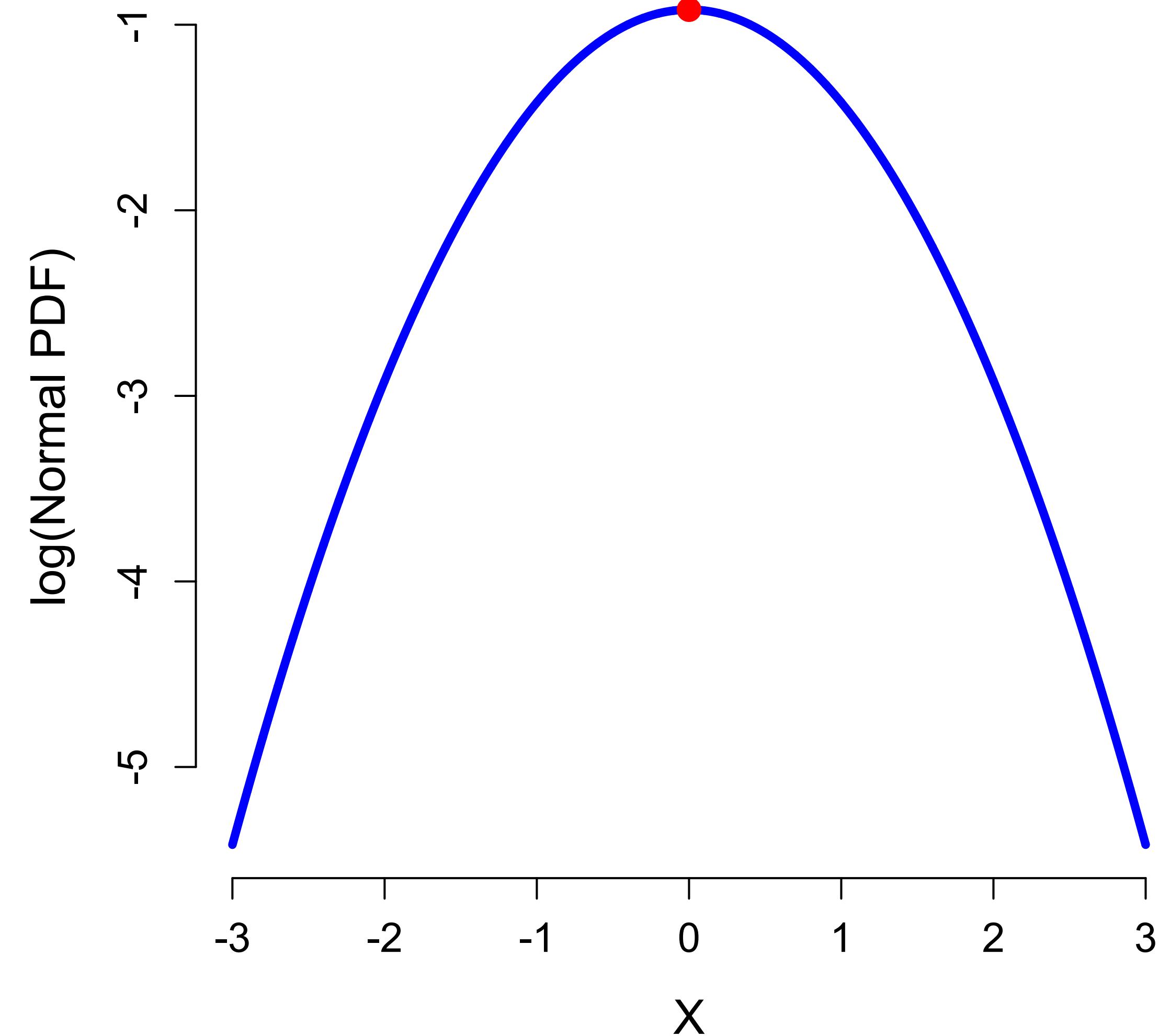
LAPLACE APPROXIMATION

- ▶ This is a general method for approximating the shape of any curve
- ▶ Because of the central limit theorem, many joint posteriors are approximately multivariate normal distributions
- ▶ The log of a gaussian is a parabola



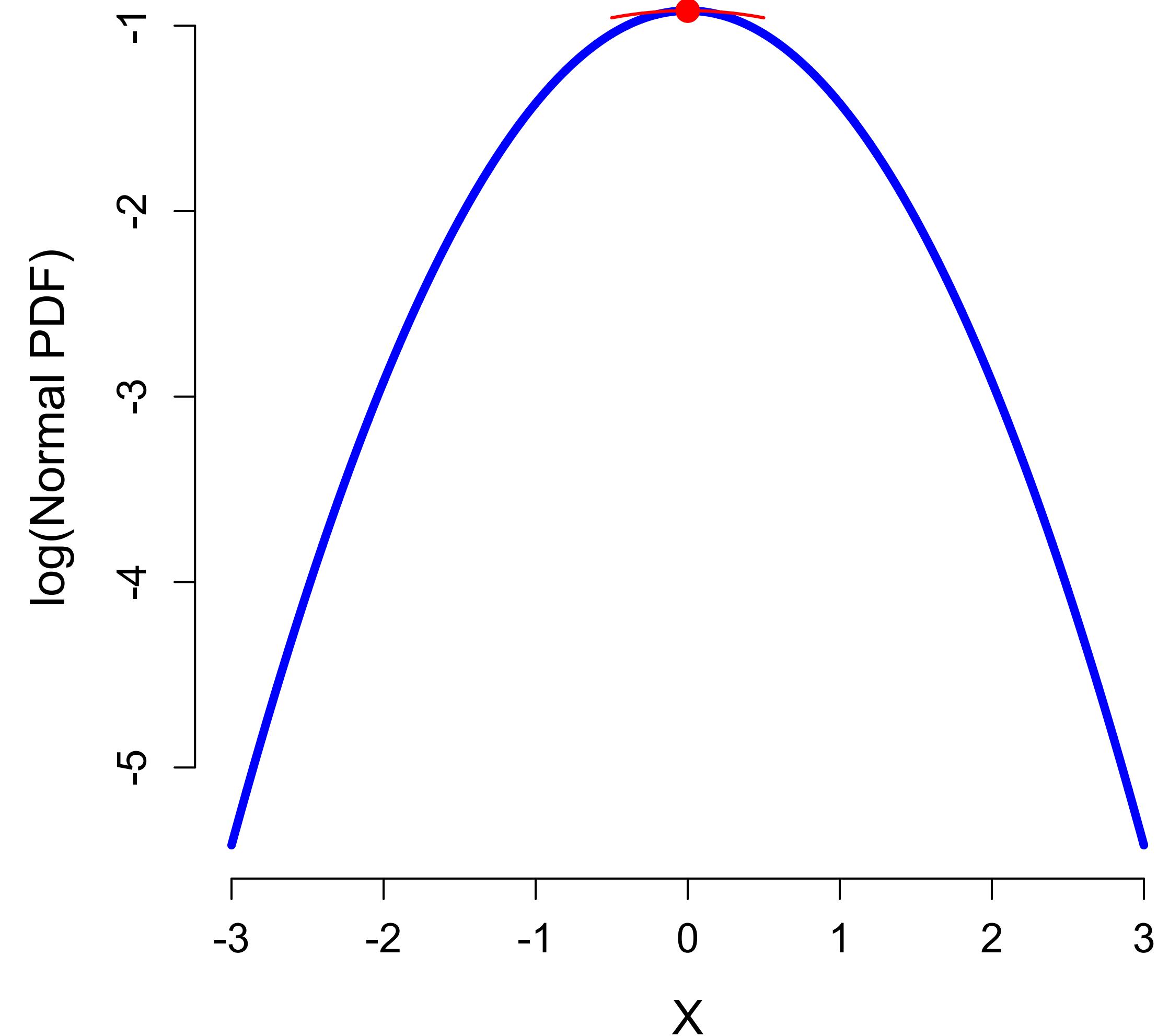
LAPLACE APPROXIMATION

- ▶ LA first finds the maximum of the log posterior
(maximum a posteriori inference)



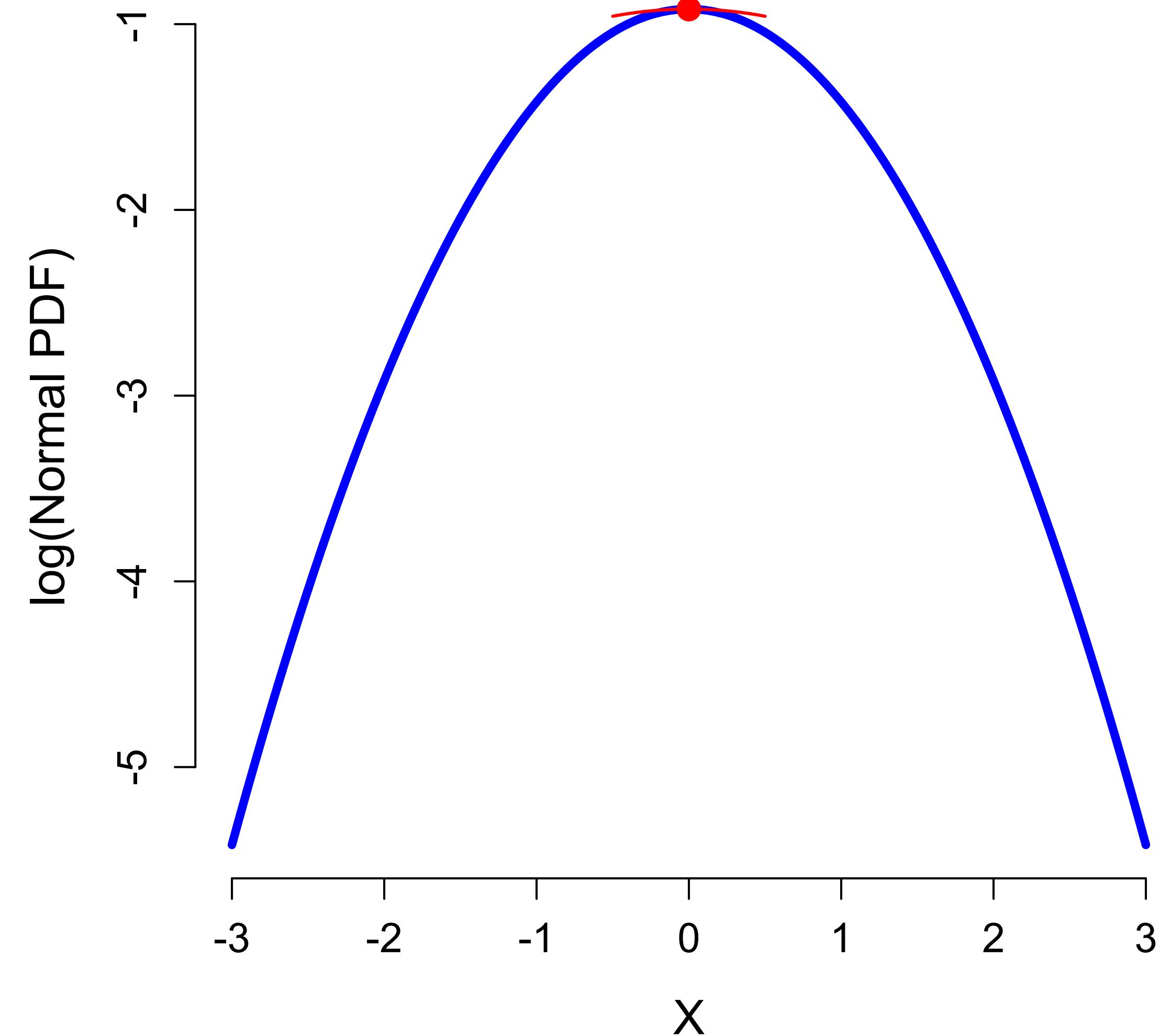
LAPLACE APPROXIMATION

- ▶ LA first finds the maximum of the log posterior (maximum a posteriori inference)
- ▶ We then estimate the **hessian matrix** to approximate the curvature at the maximum



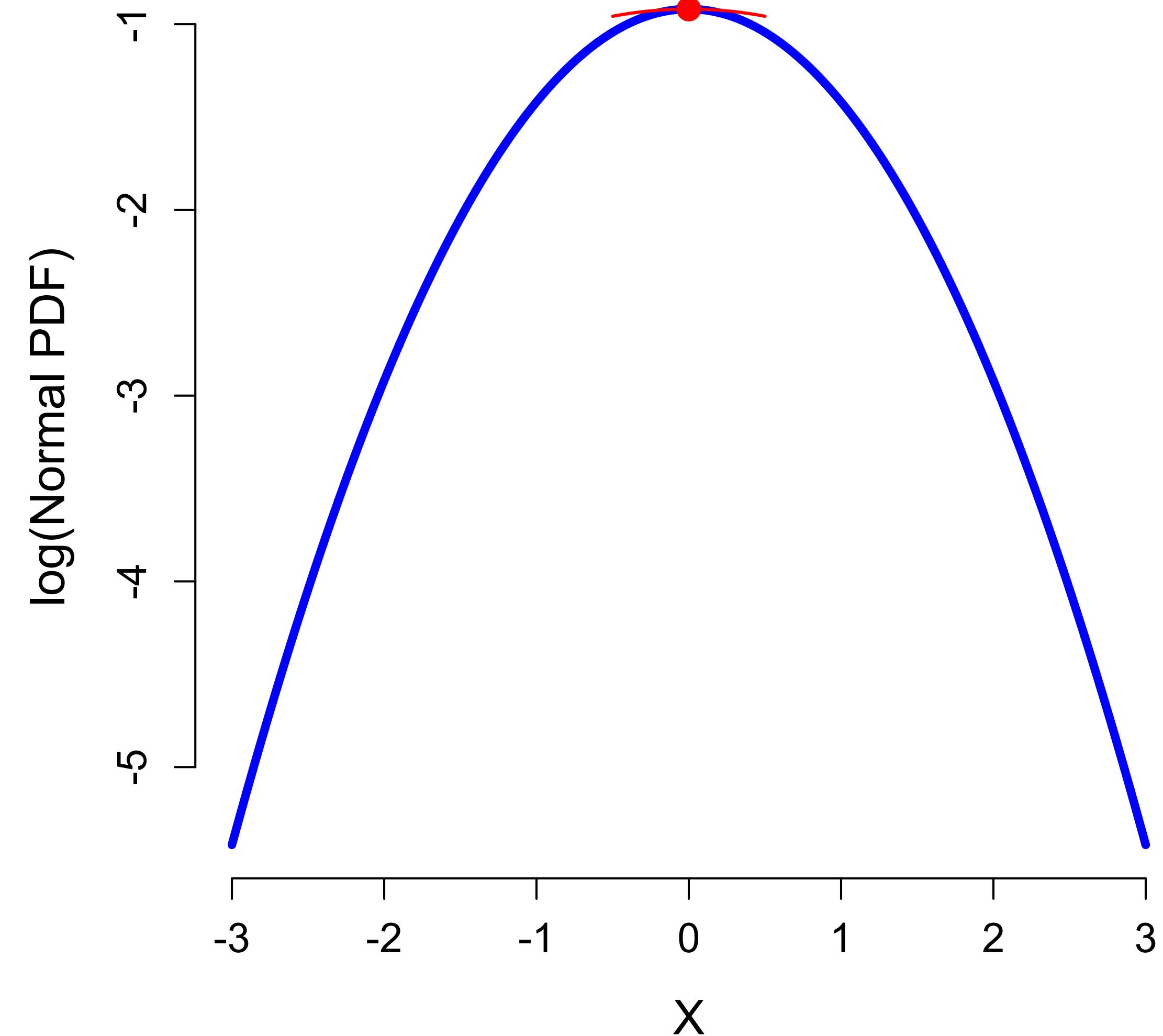
LAPLACE APPROXIMATION

- ▶ LA first finds the maximum of the log posterior (maximum a posteriori inference)
- ▶ We then estimate the **hessian matrix** to approximate the curvature at the maximum
- ▶ The hessian can be solved to estimate a **variance-covariance matrix** for all parameters



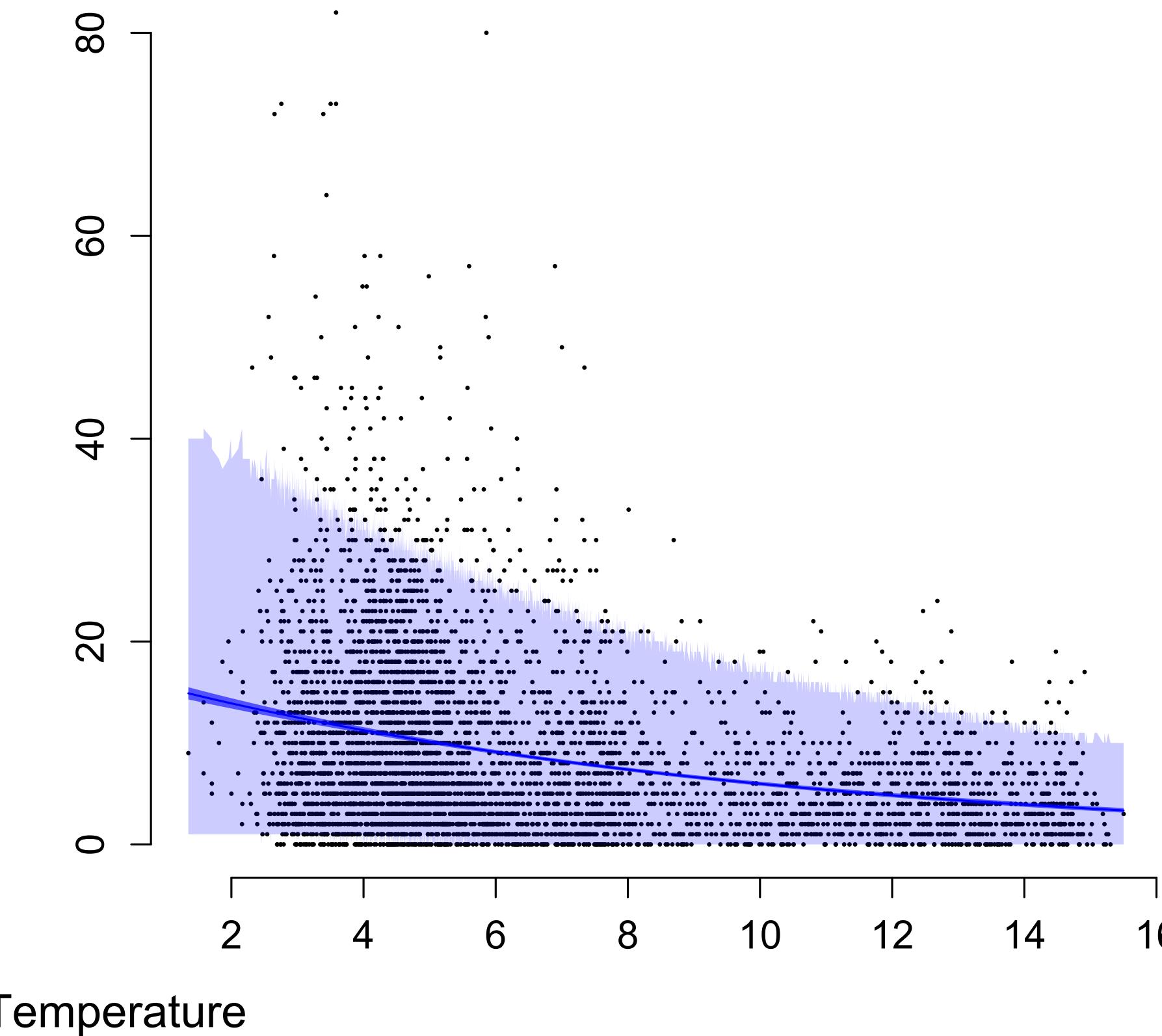
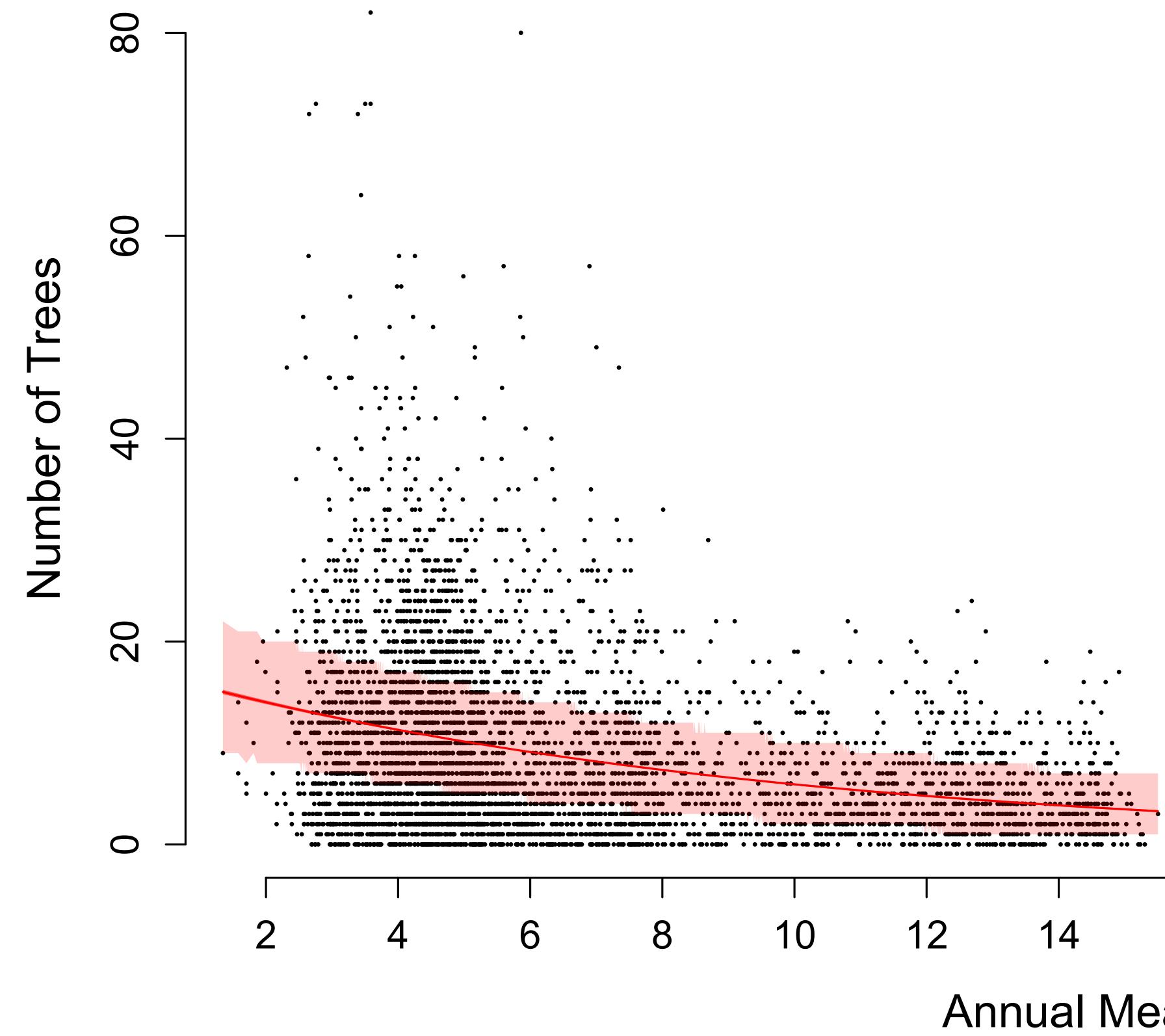
LAPLACE APPROXIMATION

- ▶ LA first finds the maximum of the log posterior (maximum a posteriori inference)
- ▶ We then estimate the **hessian matrix** to approximate the curvature at the maximum
- ▶ The hessian can be solved to estimate a **variance-covariance matrix** for all parameters
- ▶ Finally, we draw samples from a multivariate normal distribution to make posterior inference



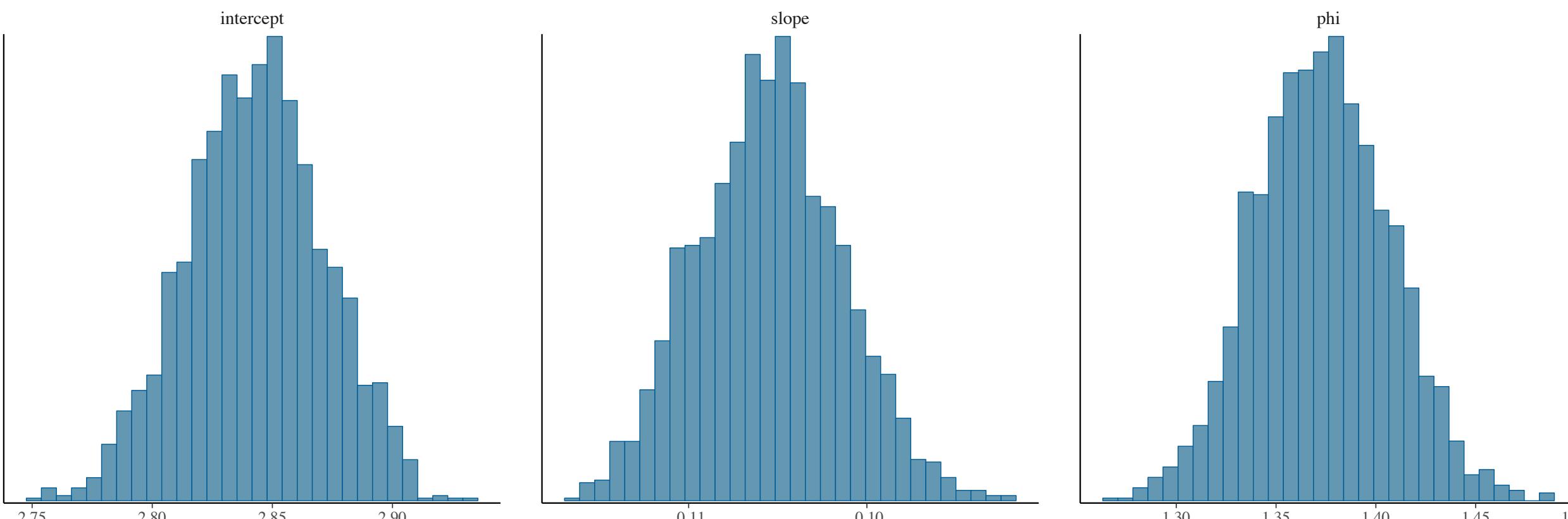
NEGATIVE BINOMIAL REGRESSION

- From the tree abundance regression, we saw a negative binomial model did best modelling both the mean and variance



NEGATIVE BINOMIAL REGRESSION

- ▶ This model has three parameters: a slope, an intercept, and a dispersion parameter
- ▶ However, phi is slightly skewed



```

log_posterior <- function(params, data) {
  intercept <- params[1]
  slope <- params[2]
  phi <- params[3]

  if(phi <= 0) return(-Inf)

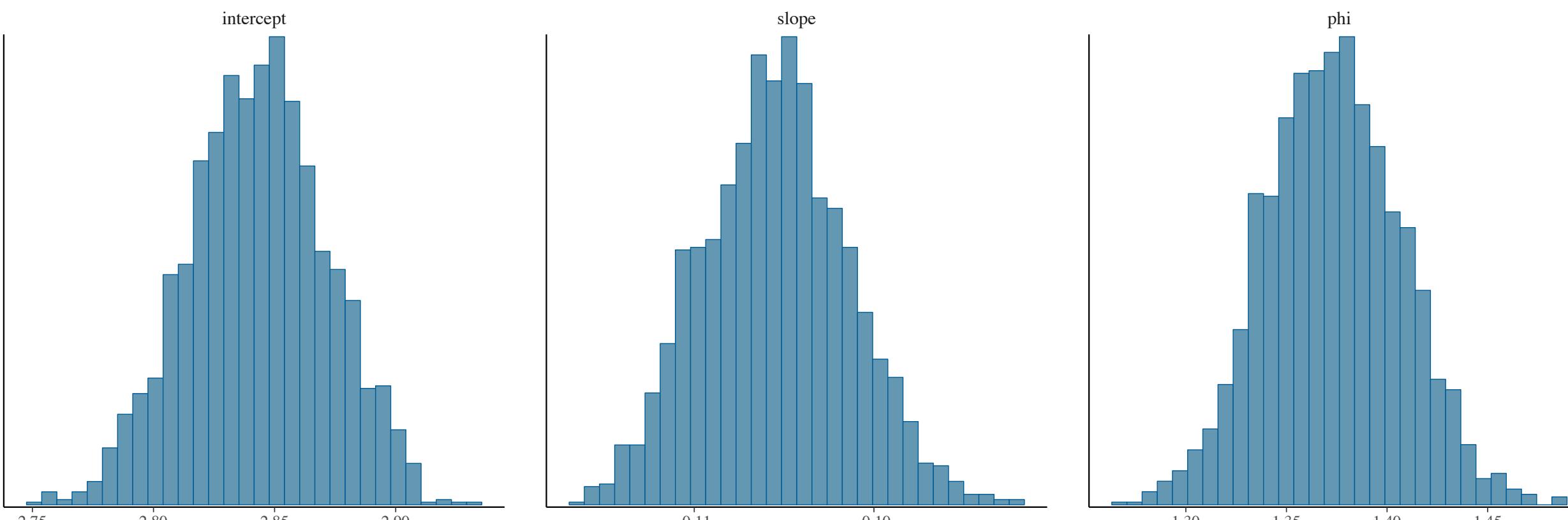
  # using a log link function, hence exp()
  mu <- exp(intercept + slope *
    data$annual_mean_temp)

  loglik <- dnbinom(data$n, mu = mu, size = phi,
    log=TRUE)
  logprior <- dnorm(intercept, 0, 10, log = TRUE) +
    dnorm(slope, 0, 15, log=TRUE) +
    dexp(phi, 0.1, log=TRUE)
  return(loglik + logprior)
}

```

NEGATIVE BINOMIAL REGRESSION

- ▶ Correct phi by doing estimation on the log of phi
- ▶ This also removes the need to check that $\phi \geq 0$



```

log_posterior <- function(params, data) {
  intercept <- params[1]
  slope <- params[2]
  phi <- exp(params[3])

  # using a log link function, hence exp()
  mu <- exp(intercept + slope *
    data$annual_mean_temp)

  loglik <- dnb(nom(data$n, mu = mu, size = phi,
    log=TRUE)
  logprior <- dnorm(intercept, 0, 10, log = TRUE) +
    dnorm(slope, 0, 15, log=TRUE) +
    dexp(phi, 0.1, log=TRUE)
  return(loglik + logprior)
}
  
```

NEGATIVE BINOMIAL REGRESSION

- ▶ Then it is simply a matter of using `optim` with `hessian=TRUE`
- ▶ MAP estimate is nearly identical to the means from Stan

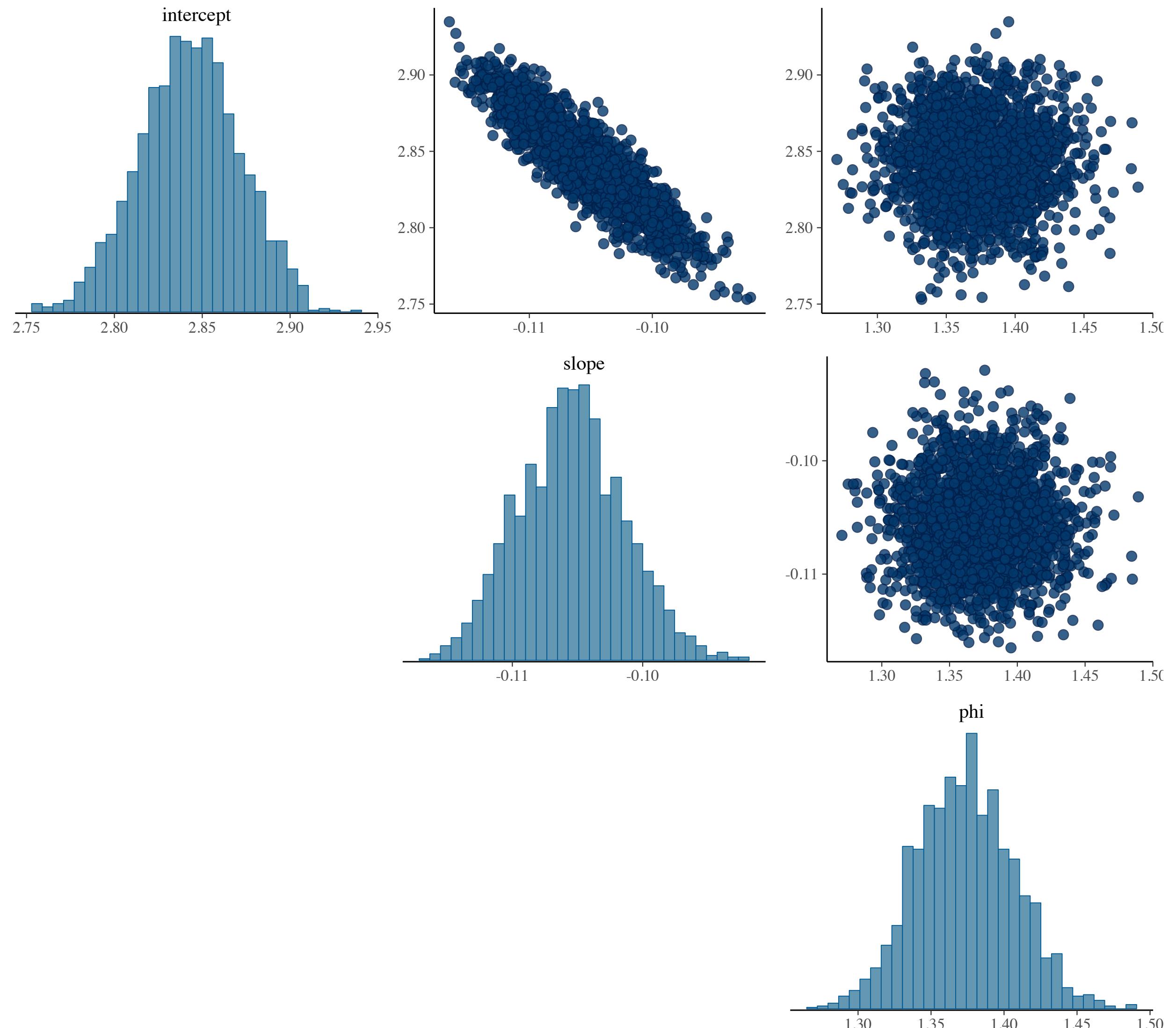
```
initial_values <- c(0, 0, 1)
fit <- optim(initial_values, log_posterior,
control = list(fnscale = -1), hessian = TRUE,
data = dat)

fit$par ## remember to exp() the last parameter
colMeans(samples_nb)
```

	Stan	Laplace
intercept	2.842872	2.8419241
slope	-0.1054756	-0.1054590
phi	1.373309	1.372184

NEGATIVE BINOMIAL REGRESSION

- ▶ We can see from Stan that slope and intercept are strongly correlated, phi less so
- ▶ (use `mcmc_pairs` for this plot)



NEGATIVE BINOMIAL REGRESSION

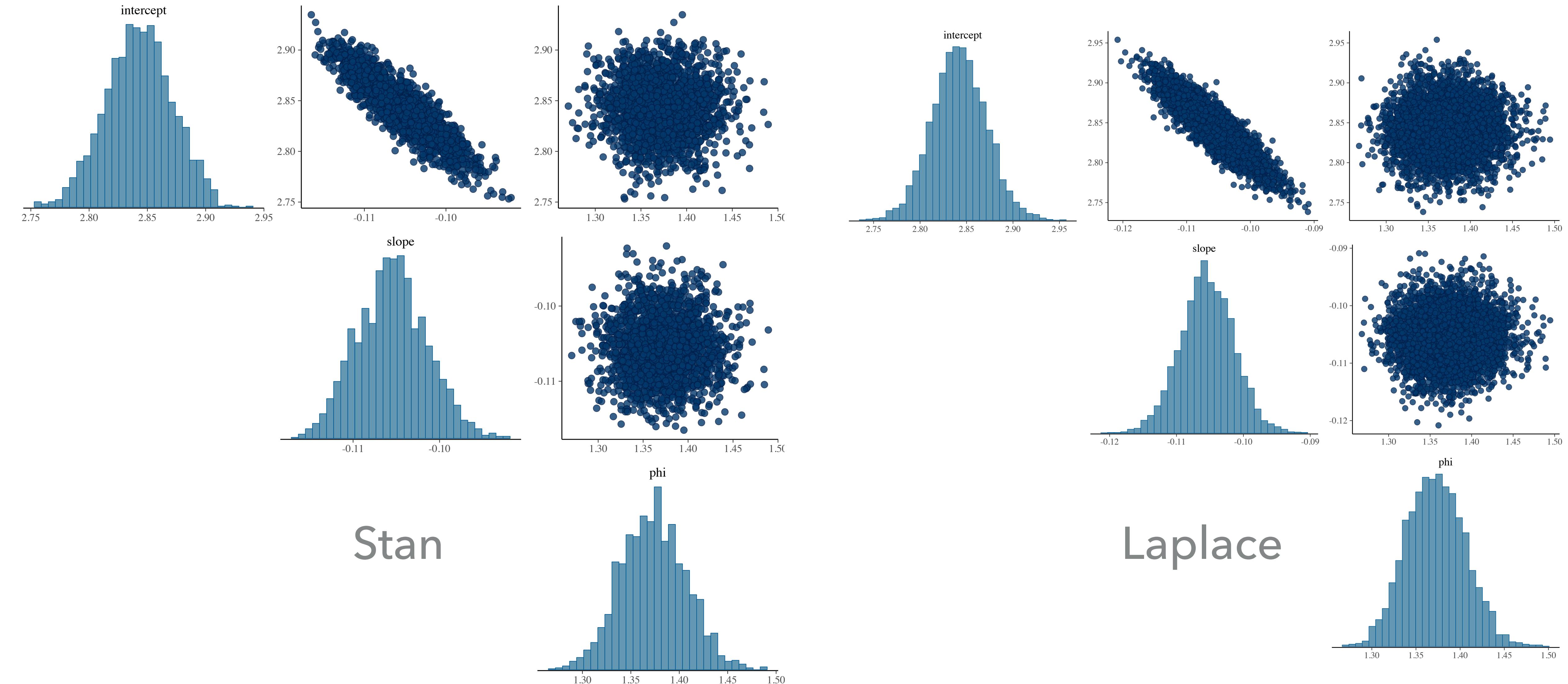
- ▶ Next step is to solve the hessian for the variance covariance matrix
- ▶ Finally, we can draw posterior samples and compare to Stan

```
vcv_mat <- solve(-fit$hessian)

samples_nb_la <- rmvnorm(5000, fit$par, vcv_mat)
colnames(samples_nb_la) <-
  c("intercept", "slope", "phi")

mcmc_pairs(samples_nb_la)
```

NEGATIVE BINOMIAL REGRESSION



Stan

Laplace

NEGATIVE BINOMIAL REGRESSION

