Bayesian Data Analysis Demos 3.1, 3.2, 3.3, 3.4

Hendra Bunyamin

2021-09-02

Normal model with unknown mean and variance (BDA3 section 3.2 on p. 64).

We demonstrate this list of distributions,
1. Multivariate joint distribution,
2. Conditional distribution,
3. Marginal distribution,
4. Marginalization, and
5. Posterior predictive distribution.

Particularly, `ggplot2`, ‘grid, and gridExtra are used for plotting, tidyr for manipulating data frames

```r
library(ggplot2)
theme_set(theme_minimal())
library(grid)
library(gridExtra)
library(tidyr)
```

Generic part common for Demos 3.1-3.4

Our dataset (y) is as follows:

```r
y <- c(93, 112, 122, 135, 122, 150, 118, 90, 124, 114)
```

Let’s compute the sufficient statistics from dataset,

$$\begin{aligned} n &= \text{number of instances in our dataset} && (1) \\ s^2 &= \text{sample variance of our dataset} && (2) \\ \overline{y} &= \text{sample mean of our dataset} && (3) \end{aligned}$$

as follows:

```r
n <- length(y)      # 10
s2 <- var(y)        # 315.7778 dan sigma = 17.77014
mean_y <- mean(y)   # 118
```

Factorize the joint posterior distribution $\Pr(\mu, \sigma^2 \mid y)$ as follows

$$\Pr(\mu, \sigma^2 \mid y) = \Pr(\sigma^2 \mid y) \times \Pr(\mu \mid \sigma^2, y). \tag{4}$$

Recall that the marginal distribution $\Pr(\sigma^2 \mid y)$ is explained in Eq. (3.5) from the book,

$$\Pr(\sigma^2 \mid y) \propto (\sigma^2)^{-(n+1)/2} \exp\left(-\frac{(n-1)s^2}{2\sigma^2}\right) \tag{5}$$

which is a scaled inverse-χ^2 density:

$$\sigma^2 \mid y \sim \text{Inv-}\chi^2(n-1, s^2). \tag{6}$$

Specifically, the code for marginal density function of Inv-$\chi^2(n-1, s^2)$ with $n-1 =$ `nu` and $s^2 =$ `s2` is

```
# helper functions to evaluate
# scaled inverse chi-squared distribution
# d = probability density function of the distribution
dsinvchisq <- function(x, nu, s2){
  exp(log(nu/2)*nu/2 - lgamma(nu/2) + log(s2)/2*nu - log(x)*(nu/2+1) - (nu*s2/2)/x)
}
```

just like described on page 576-577 in *Table A.1* in *Appendix A* of the book .

Furthermore, the marginal distribution of $\sigma^2 \mid y$ has a remarkable similarity to the analogous sampling theory result: conditional on σ^2 (and μ), the distribution of the appropriately scaled sufficient statistic,

$$\frac{(n-1)s^2}{\sigma^2} \sim \chi^2_{n-1} \iff \sigma^2 \sim \frac{(n-1)s^2}{\chi^2_{n-1}} \tag{7}$$

By utilizing Eq. (7), we can sample the values of σ^2 with $n-1 =$ `n_minus_1`, $s^2 =$ `s2`, and #samples = `n` as follows:

```
# helper functions to sample from
# scaled inverse chi-squared distribution
# r = random sampling from the distribution
rsinvchisq <- function(n, n_minus_1, s2, ...) n_minus_1*s2 / rchisq(n, n_minus_1, ...)
```

Recall that the conditional posterior $\Pr(\mu \mid \sigma^2, y)$ is explained in Eq. (3.3) from the book,

$$\mu \mid \sigma^2, y \sim \mathrm{N}(\overline{y}, \sigma^2/n). \tag{8}$$

Sampling from the Joint Posterior Distribution (page 64-65)

Now let's sample from the joint posterior distribution using the factorization in Eq. (4).

Firstly, we sample 1000 random numbers from marginal posterior distribution, $\Pr(\sigma^2 \mid y)$.

```
num_samples <- 1000
sigma2 <- rsinvchisq(num_samples, n-1, s2)
```

Secondly, we sample from conditional posterior distribution, $\Pr(\mu \mid \sigma^2, y)$ as follows:

```
mu <- mean_y + sqrt(sigma2/n)*rnorm(length(sigma2))
```

Note that `rnorm` generates a sample of standard normal distribution. Therefore, now `mu` (μ) shall have *mean*=$\overline{y}$ and *variance*=σ^2/n which is what we want from Eq. (8).

Now that we have μ and σ^2 from y, we can sample from predictive distribution $\Pr(\tilde{y} \mid y)$ (page 66) that is

$$\tilde{y} \sim \mathrm{N}(\mu, \sigma^2). \tag{9}$$

```
# apply square root of sigma to obtain sigma
sigma <- sqrt(sigma2)
# generate a sample of y
ynew <- rnorm(num_samples, mu, sigma)
```

Compute the Density in a Grid

For μ (`mu`), σ (`sigma`) and $\tilde{y}$ (`ynew`), compute the density in a grid ranges for the grids.

```
t1l <- c(90, 150)    # for mu,   mu = 118
t2l <- c(10, 60)     # for sigma, sigma = 17.77014
nl <- c(50, 185)     # for ynew
t1 <- seq(t1l[1], t1l[2], length.out = num_samples)
t2 <- seq(t2l[1], t2l[2], length.out = num_samples)
xynew <- seq(nl[1], nl[2], length.out = num_samples)
```

Let's compute the exact marginal posterior density of mu (page 66).
Recall that

$$z \mid y \sim t_{n-1} \tag{10}$$

with

$$z = \frac{\mu - \overline{y}}{s/\sqrt{n}}. \tag{11}$$

However, we need to transform Eq. (10) into

$$\mu \mid y \sim t_{n-1}(?,?) \text{ with } \mu = \overline{y} + z\frac{s}{\sqrt{n}}. \tag{12}$$

Therefore, we need to transform z into μ. Remember that transforming $\Pr(z \mid y)$ to $\Pr(\mu \mid y)$ requires $|J|$, Jacobian that is $|\frac{dz}{d\mu}|$. Specifically,

$$\mu = \overline{y} + z\frac{s}{\sqrt{n}} \iff z = \frac{\mu - \overline{y}}{s/\sqrt{n}} \iff |\frac{dz}{d\mu}| = \frac{1}{s/\sqrt{n}} \tag{13}$$

Hence, the transformation result shall be

$$\Pr(\mu \mid y) = \Pr(z \mid y) \times |\frac{dz}{d\mu}| = \Pr(\frac{\mu - \overline{y}}{s/\sqrt{n}} \mid y) \times \frac{1}{s/\sqrt{n}} \tag{14}$$

and coded in R as follows:

```
# multiplication by 1./sqrt(s2/n) is due to the transformation of
# variable z=(x-mean(y))/sqrt(s2/n), see BDA3 p. 21, the formula is on page 66
posterior_marginal_mu <- dt((t1-mean_y) / sqrt(s2/n), n-1) / sqrt(s2/n)
```

Estimate the marginal density using samples and ad hoc Gaussian kernel approximation. *Ad hoc* means that is done only when needed for a specific purpose, without planning or preparation.

```
posterior_marginal_mu_kernel <- density(mu, adjust = 2, n = num_samples, from = t1l[1], to = t1l[2])$y
```

Compute the exact marginal density of sigma (page 65). Previously, we have $z \mid y \sim \text{Inv-}\chi^2(n-1, s^2)$ with $z = \sigma^2$. Hence, we want to transform z into σ with transformation: $\sigma = \sqrt{z}$. Remember that transforming $\Pr(z \mid y)$ to $\Pr(\sigma \mid y)$ requires $|J|$, Jacobian that is $|\frac{dz}{d\sigma}|$. Specifically,

$$\sigma = \sqrt{z} \iff z = (\sigma)^2 \iff |\frac{dz}{d\sigma}| = 2 * \sigma \tag{15}$$

Moreover, the density function of $\sigma \mid y$ is defined as

$$\Pr(\sigma \mid y) = \Pr(z \mid y) \times |\frac{dz}{d\sigma}| = \Pr((\sigma)^2 \mid y) \times 2 * \sigma. \tag{16}$$

In the code below, `t2` denotes σ.

```
# the multiplication by 2*t2 is due to the transformation of
# variable z=t2^2, see BDA3 p. 21
posterior_marginal_sigma <- dsinvchisq(t2^2, n-1, s2) * 2*t2
```

Similarly, let's estimate the marginal density using samples and ad hoc Gaussian kernel approximation

```
posterior_marginal_sigma_kernel <- density(sigma, n = num_samples, from = t2l[1], to = t2l[2])$y
```

Let's compute the exact posterior predictive density.

```
# multiplication by 1./sqrt(s2/n) is due to the transformation of variable
# see BDA3 p. 21
posterior_new_predictive <- dt((xynew - mean_y ) / sqrt(s2*(1+1/n)), n-1) / sqrt(s2*(1+1/n))
```

Let's evaluate the joint posterior density in a grid. Note that the following is not normalized, but for plotting contours it does not matter. The joint posterior density is computed by Eq. (4).

```
# Combine grid points into another data frame
# with all pairwise combinations
dfj <- data.frame(t1 = rep(t1, each = length(t2)),
                  t2 = rep(t2, length(t1)))
# Here is the exact joint posterior computation based on Eq. (4) but using a standard deviation, not a
dfj$z <- dsinvchisq(dfj$t2^2, n-1, s2) * 2*dfj$t2 * dnorm(dfj$t1, mean_y, dfj$t2/sqrt(n))
# breaks for plotting the contours
cl <- seq(1e-5, max(dfj$z), length.out = 6)
```

Demo 3.1 Visualise the joint and marginal densities

Visualise the joint density and marginal densities of the posterior of normal distribution with unknown mean and variance.

Create a plot of the marginal density of mu,

$$\Pr(\mu \mid y).$$

```
dfm <- data.frame(t1, Exact = posterior_marginal_mu, Empirical = posterior_marginal_mu_kernel) %>% gathe
margmu <- ggplot(dfm) +
  geom_line(aes(t1, p, color = grp)) +
  coord_cartesian(xlim = t1l) +
  labs(title = 'Marginal of mu', x = '', y = '') +
  scale_y_continuous(breaks = NULL) +
  theme(legend.background = element_blank(),
        legend.position = c(0.75, 0.8),
        legend.title = element_blank())
```

Create a plot of the marginal density of sigma,

$$\Pr(\sigma \mid y).$$

```
dfs <- data.frame(t2, Exact = posterior_marginal_sigma, Empirical = posterior_marginal_sigma_kernel) %>
margsig <- ggplot(dfs) +
  geom_line(aes(t2, p, color = grp)) +
  coord_cartesian(xlim = t2l) +
  coord_flip() +
  labs(title = 'Marginal of sigma', x = '', y = '') +
  scale_y_continuous(breaks = NULL) +
  theme(legend.background = element_blank(),
        legend.position = c(0.75, 0.8),
        legend.title = element_blank())
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```

Create a plot of the joint posterior density,

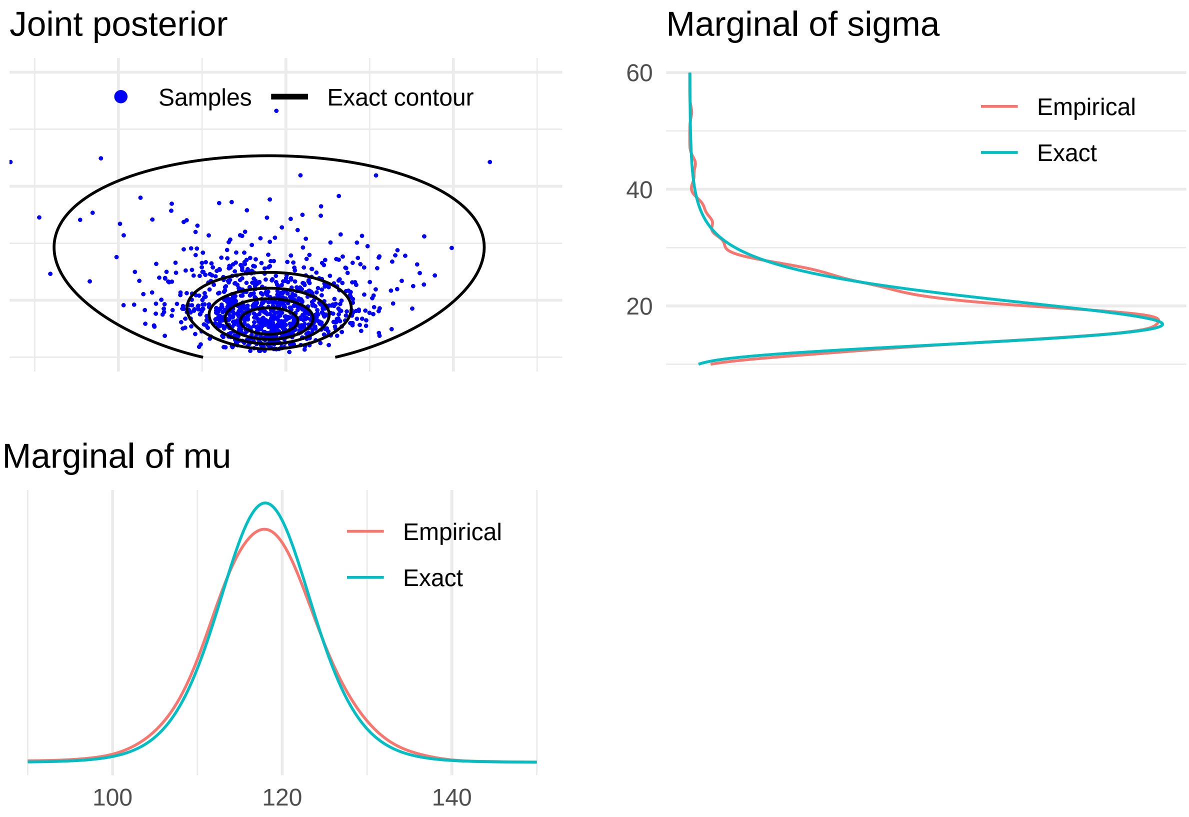
$$\Pr(\mu, \sigma \mid y).$$

```
joint1labs <- c('Samples','Exact contour')
joint1 <- ggplot() +
  geom_point(data = data.frame(mu,sigma), aes(mu, sigma, col = '1'), size = 0.1) +
  geom_contour(data = dfj, aes(t1, t2, z = z, col = '2'), breaks = cl) +
  coord_cartesian(xlim = t1l,ylim = t2l) +
  labs(title = 'Joint posterior', x = '', y = '') +
  scale_y_continuous(labels = NULL) +
  scale_x_continuous(labels = NULL) +
  scale_color_manual(values=c('blue', 'black'), labels = joint1labs) +
  guides(color = guide_legend(nrow  = 1, override.aes = list(
    shape = c(16, NA), linetype = c(0, 1), size = c(2, 1)))) +
  theme(legend.background = element_blank(),
        legend.position = c(0.5, 0.9),
        legend.title = element_blank())
```

Combine the plots

```
# blank plot for combining the plots
bp <- grid.rect(gp = gpar(col = 'white'))
```

```
grid.arrange(joint1, margsig, margmu, bp, nrow = 2)
```



Demo 3.2 Visualise factored distribution

Visualise factored sampling and the corresponding marginal and conditional densities.

Create another plot of the joint posterior

```
# data frame for the conditional of mu and marginal of sigma
dfc <- data.frame(mu = t1, marg = rep(sigma[1], length(t1)),
                   cond = sigma[1] + dnorm(t1 ,mean_y, sqrt(sigma2[1]/n)) * 100) %>%
  gather(grp, p, marg, cond)
# legend labels for the following plot
joint2labs <- c('Exact contour plot', 'Sample from joint post.',
               'Cond. distribution of mu', 'Sample from the marg. of sigma')
joint2 <- ggplot() +
  geom_contour(data = dfj, aes(t1, t2, z = z, col = '1'), breaks = cl) +
  geom_point(data = data.frame(m = mu[1], s = sigma[1]), aes(m , s, color = '2')) +
  geom_line(data = dfc, aes(mu, p, color = grp), linetype = 'dashed') +
  coord_cartesian(xlim = t1l,ylim = t2l) +
  labs(title = 'Joint posterior', x = '', y = '') +
  scale_x_continuous(labels = NULL) +
  scale_color_manual(values=c('blue', 'darkgreen','darkgreen','black'), labels = joint2labs) +
  guides(color = guide_legend(nrow  = 2, override.aes = list(
    shape = c(NA, 16, NA, NA), linetype = c(1, 0, 1, 1)))) +
  theme(legend.background = element_blank(),
        legend.position = c(0.5, 0.85),
        legend.title = element_blank())
```

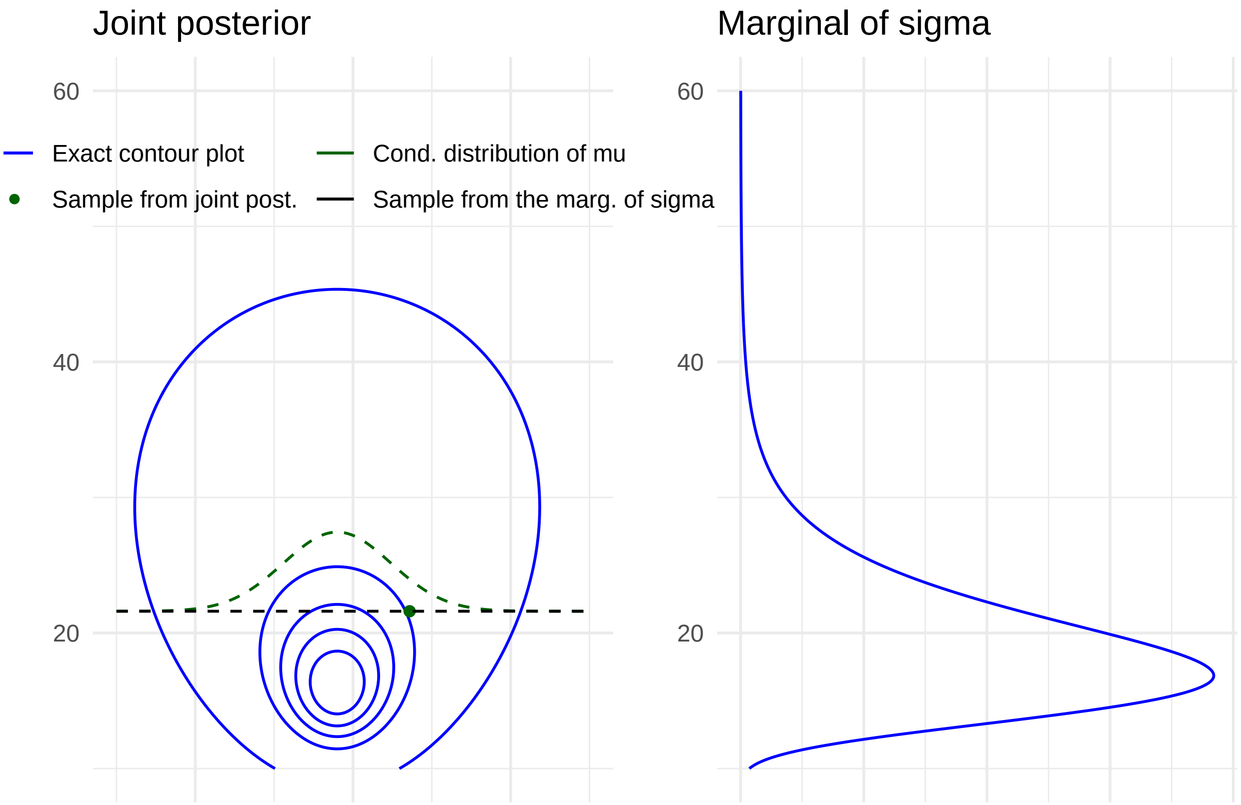
Create another plot of the marginal density of sigma

```
margsig2 <- ggplot(data = data.frame(t2, posterior_marginal_sigma)) +
  geom_line(aes(t2, posterior_marginal_sigma), color = 'blue') +
  coord_cartesian(xlim = t2l) +
  coord_flip() +
  labs(title = 'Marginal of sigma', x = '', y = '') +
  scale_y_continuous(labels = NULL)
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```

Combine the plots

```
grid.arrange(joint2, margsig2, ncol = 2)
```



Demo 3.3 Visualise the marginal distribution of mu

Visualise the marginal distribution of mu as a mixture of normals.

Calculate conditional pdfs for each sample

```
condpdfs <- sapply(t1, function(x) dnorm(x, mean_y, sqrt(sigma2/n)))
```

Create a plot of some of them

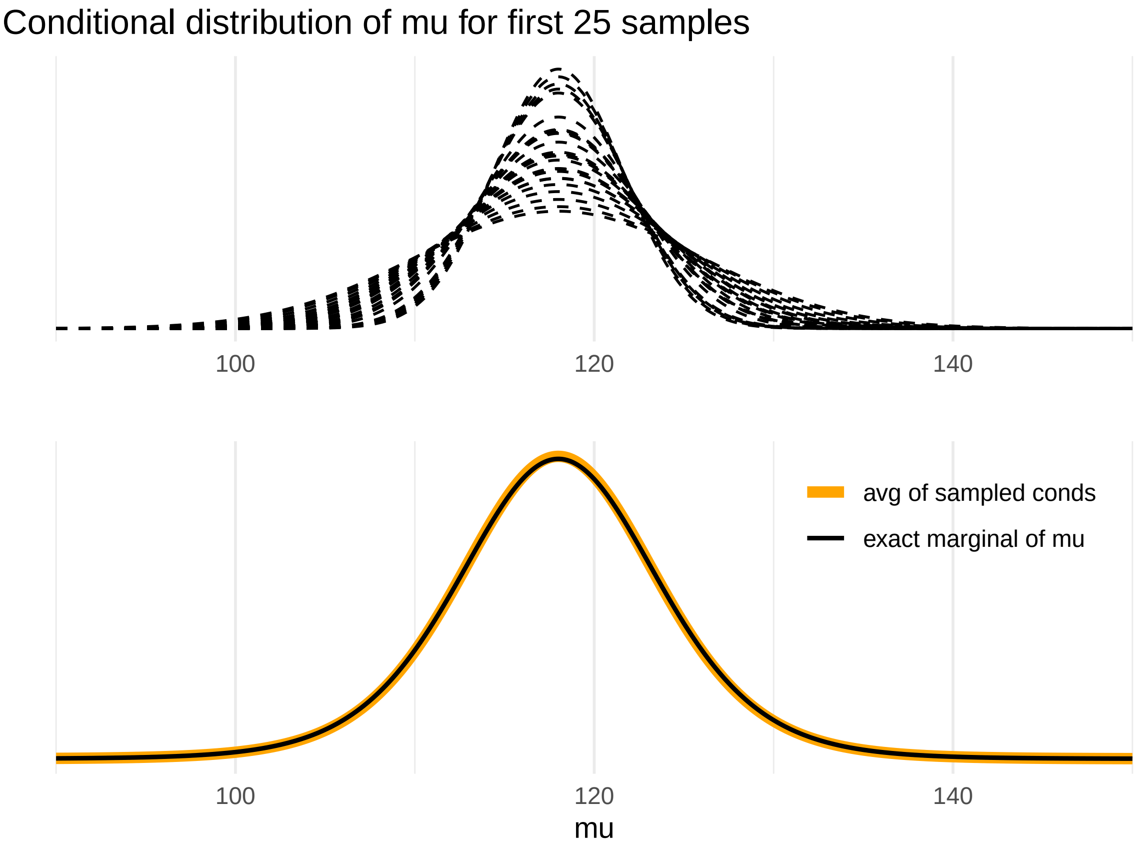
```
# data frame of 25 first samples
dfm25 <- data.frame(t1, t(condpdfs[1:25,])) %>% gather(grp, p, -t1)
condmu <- ggplot(data = dfm25) +
  geom_line(aes(t1, p, group = grp), linetype = 'dashed') +
  labs(title = 'Conditional distribution of mu for first 25 samples', y = '', x = '') +
  scale_y_continuous(breaks = NULL)
```

Create a plot of their mean

```
dfsam <- data.frame(t1, colMeans(condpdfs), posterior_marginal_mu) %>% gather(grp,p,-t1)
# labels
mulabs <- c('avg of sampled conds', 'exact marginal of mu')
meanmu <- ggplot(data = dfsam) +
  geom_line(aes(t1, p, size = grp, color = grp)) +
  labs(y = '', x = 'mu') +
  scale_y_continuous(breaks = NULL) +
  scale_size_manual(values = c(2, 0.8), labels = mulabs) +
  scale_color_manual(values = c('orange', 'black'), labels = mulabs) +
  theme(legend.position = c(0.8, 0.8),
        legend.background = element_blank(),
        legend.title = element_blank())
```

Combine the plots

```
grid.arrange(condmu, meanmu, ncol = 1)
```



Demo 3.4 Visualise posterior predictive distribution.

Visualise sampling from the posterior predictive distribution. Calculate each predictive pdf with given mu and sigma.

```
ynewdists <- sapply(xynew, function(x) dnorm(x, mu, sigma))
```

Create plot of the joint posterior with a draw from the posterior predictive distribution, highlight the first sample create a plot of the joint density

```
# data frame of dirst draw from sample the predictive along with the exact value for plotting
dfp <- data.frame(xynew, draw = ynewdists[1,], exact = posterior_new_predictive)
# data frame for plotting the samples
dfy <- data.frame(ynew, p = 0.02*max(ynewdists))
# legend labels
pred1labs <- c('Sample from the predictive distribution', 'Predictive distribution given the posterior
pred2labs <- c('Samples from the predictive distribution', 'Exact predictive distribution')
joint3labs <- c('Samples', 'Exact contour')
joint3 <- ggplot() +
  geom_point(data = data.frame(mu, sigma), aes(mu, sigma, col = '1'), size = 0.1) +
  geom_contour(data = dfj, aes(t1, t2, z = z, col = '2'), breaks = cl) +
  geom_point(data = data.frame(x = mu[1], y = sigma[1]), aes(x, y), color = 'red') +
  coord_cartesian(xlim = t1l,ylim = t2l) +
  labs(title = 'Joint posterior', x = 'mu', y = 'sigma') +
  scale_color_manual(values=c('blue', 'black'), labels = joint3labs) +
  guides(color = guide_legend(nrow  = 1, override.aes = list(
    shape = c(16, NA), linetype = c(0, 1), size = c(2, 1)))) +
  theme(legend.background = element_blank(),
        legend.position=c(0.5 ,0.9),
        legend.title = element_blank())
```

Create a plot of the predicitive distribution and the respective sample

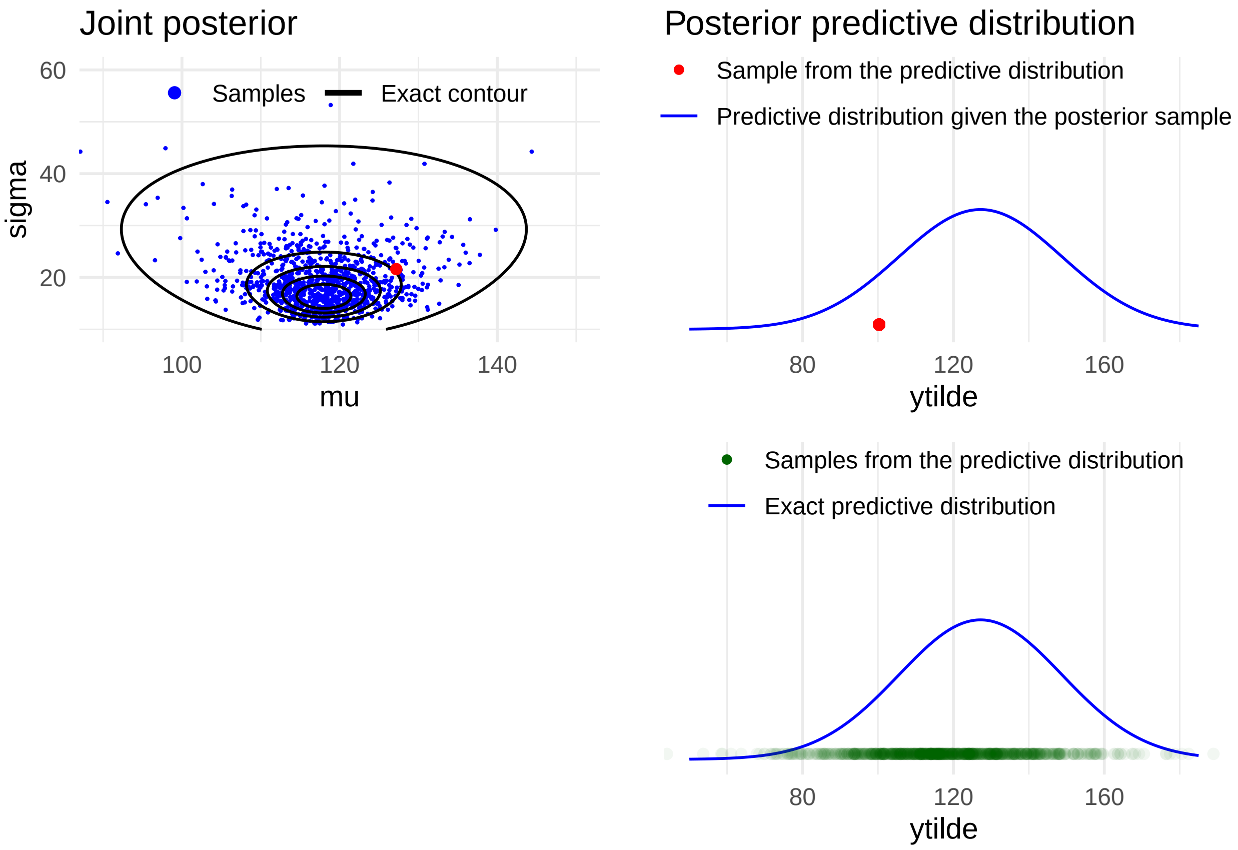
```
pred1 <- ggplot() +
  geom_line(data = dfp, aes(xynew, draw, color = '2')) +
  geom_point(data = dfy, aes(ynew[1], p, color = '1')) +
  coord_cartesian(xlim = nl, ylim = c(0,0.04)) +
  labs(title = 'Posterior predictive distribution', x = 'ytilde', y = '') +
  scale_y_continuous(breaks = NULL) +
  scale_color_manual(values = c('red', 'blue'), labels = pred1labs) +
  guides(color = guide_legend(nrow = 2, override.aes = list(
    linetype = c(0, 1), shape=c(16, NA), labels = pred1labs))) +
  theme(legend.background = element_blank(),
        legend.position = c(0.5 ,0.9),
        legend.title = element_blank())
```

Create a plot for all ynews

```
pred2 <- ggplot() +
  geom_line(data = dfp, aes(xynew, draw, color = '2')) +
  geom_point(data = dfy, aes(ynew, p, color = '1'), alpha = 0.05) +
  coord_cartesian(xlim = nl, ylim = c(0,0.04)) +
  labs(x = 'ytilde', y = '') +
  scale_y_continuous(breaks=NULL) +
  scale_color_manual(values=c('darkgreen','blue'),labels=pred2labs) +
  guides(color = guide_legend(nrow = 2, override.aes=list(
    linetype = c(0, 1), shape = c(16, NA), alpha = c(1, 1) ,labels = pred2labs))) +
  theme(legend.background = element_blank(),
        legend.position = c(0.5 ,0.9),
        legend.title = element_blank())
```

Combine the plots

```
grid.arrange(joint3, pred1, bp, pred2, nrow = 2)
```


Joint posterior
Samples
Exact contour
sigma
60
40
20
100
120
140
mu
Posterior predictive distribution
Sample from the predictive distribution
Predictive distribution given the posterior sample
80
120
160
ytilde
Samples from the predictive distribution
Exact predictive distribution
80
120
160
ytilde