BDA_A7

1. Linear model: drowning data with Stan (3 points)

The provided data drowning in the aaltobda package contains the number of people who died from drowning each year in Finland 1980–2019. A statistician is going to fit a linear model with Gaussian residual model to these data using time as the predictor and number of drownings as the target variable (see the related linear model example for the Kilpisjärvi-temperature data in the example Stan codes).
She has two objective questions:
(i) What is the trend of the number of people drowning per year? (We would plot the histogram of the slope of the linear model.)
(ii) What is the prediction for the year 2020? (We would plot the histogram of the posterior predictive distribution for the number of people drowning at $\bar{x} = 2020$.

(a)

The three mistakes in the code were:
In the model block missing a `;` after the observation model declaration.
In the parameters block should have `<lower=0>` instead of upper for sigma.
In the generated quantities block, we want to have `alpha + beta * xpred` as the first argument in `normal_rng()`, as we need to use the `xpred` for the predictive distribution.

Full corrected Stan code (including the priors determined below in section (b)(c)(d)):

```
data {
  int<lower=0> N; // number of data points
  vector[N] x; // observation year
  vector[N] y; // observation number of drowned
  real xpred; // prediction year
}
```

```
parameters {
  real alpha; // intercept
  real beta; // slope
  real<lower=0> sigma; //standard deviation is constrained to be positive
}

transformed parameters {
  vector[N] mu = alpha + beta*x; // linear model
}

model {
  alpha ~ normal(138, 50); // prior
  beta ~ normal(0, 26); // prior
  sigma ~ normal(0, 1); // as sigma is constrained to be positive,
  y ~ normal(mu, sigma); // observation model / likelihood
}

generated quantities {
  // sample from predictive distribution
  real ypred = normal_rng(alpha + beta * xpred, sigma);
}
```

(b)

In the assignment we are told that the approximate historical mean yearly number of drownings is 138, and it is very unlikely that the mean number of drownings changes more than 50% in one year. Based on this we can set σ_β so that we have $Pr(-69 < \beta < 69) = 0.99$. As can be seen below, a standard deviation of 26 is a good fit.

```
qnorm(p=c(0.005, 0.995), mean=0, sd=26)
```

```
[1] -66.97156  66.97156
```

```
qnorm(p=c(0.005, 0.995), mean = 138, sd=50)
```

```
[1]   9.208535 266.791465
```

```
qnorm(p=c(0.005, 0.995), mean = 0, sd=100)
```

```
[1] -257.5829  257.5829
```

(c)

For adding the priors to our Stan model we can either hardcode them in the stan file in the model block, or we can pass them in as part of the data (for example to make it easier to do sensitivity analysis with different priors), declaring them in the data block and using them as data variables in the model block.
Here we simply hardcoded them in the model block, adding the line: `beta ~ normal(0, 26);`

(d)

In a similar way to β we also determined a weakly informative prior for the intercept α by first calculating what standard deviation would give us reasonable values (given historical data) within a 99% interval and with mean $mu_{\alpha} = 138$. Then we also calculated what would give a similar distribution if we standardized the data and set the mean to zero. We chose $\alpha \sim normal(138, 50)$ for the first model and $\alpha \sim normal(0, 100)$ for the one with normalized data.

Model using standardized data (done in Stan code):

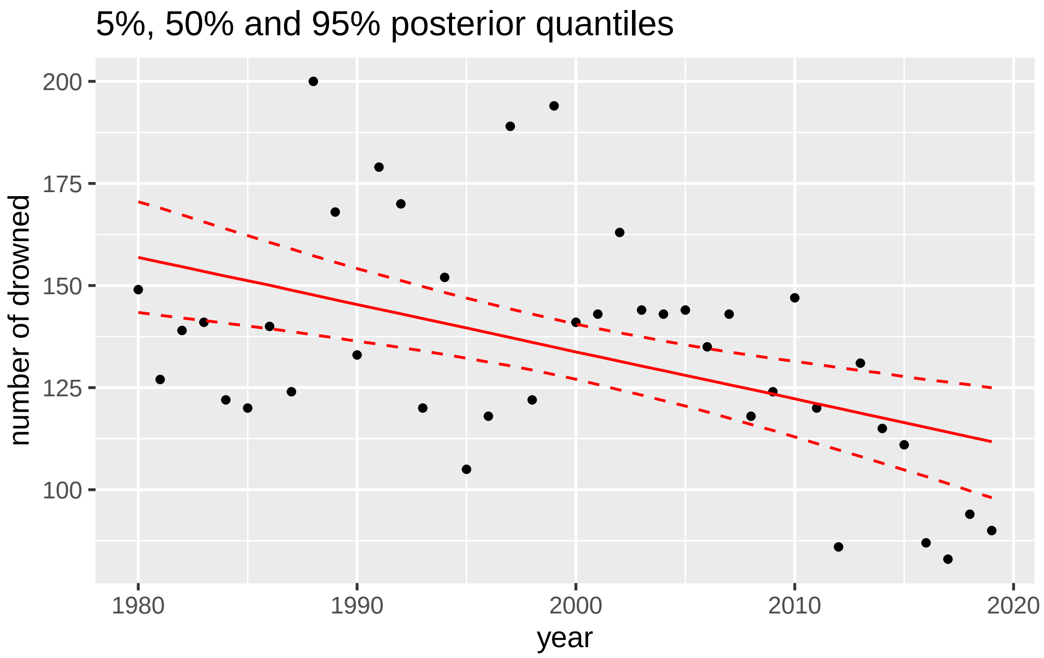
```
data {
  int<lower=0> N; // number of data points
  vector[N] x; // observation year
  vector[N] y; // observation number of drowned
  real xpred; // prediction year
}

transformed data {
  // deterministic transformations of data (std = standardized)
  vector[N] x_std = (x - mean(x)) / sd(x);
  vector[N] y_std = (y - mean(y)) / sd(y);
  real xpred_std = (xpred - mean(x)) / sd(x);
}

parameters {
  real alpha; // intercept
  real beta; // slope
  real<lower=0> sigma_std; //standard deviation is constrained to be positive
}
```

```
transformed parameters {
  vector[N] mu_std = alpha + beta * x_std; // linear model
}

model {
  alpha ~ normal(0, 100); // prior
  beta ~ normal(0, 26); // prior
  sigma_std ~ normal(0, 1); // as sigma is constrained to be positive,
  y_std ~ normal(mu_std, sigma_std); // observation model / likelihood
}

generated quantities {
  // transform to the original data scale
  vector[N] mu = mu_std * sd(y) + mean(y);
  real<lower=0> sigma = sigma_std * sd(y);
  // sample from the predictive distribution
  real ypred = normal_rng((alpha + beta * xpred_std) * sd(y) + mean(y),
                           sigma_std * sd(y));
}
```

```r
mod2 <- cmdstan_model(stan_file = "A7_drownings_stand.stan")
fit2 <- mod2$sample(data = drowning_data, seed = SEED, refresh=1000)
draws2 <- fit2$draws(format="df")
```

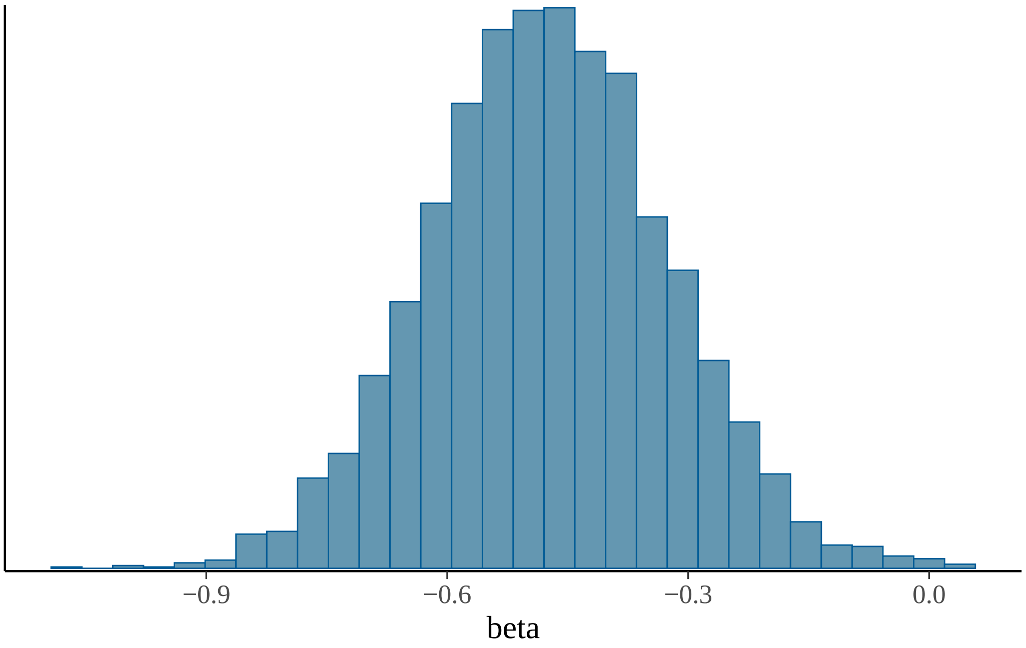
Plots using the standardized data and model:

```r
# modified from cmdStanR demo 5.1 and 5.2
mu <- draws2 |>
  as_tibble() |>
  select(starts_with("mu[")) |>
  apply(2, quantile, c(0.05, 0.5, 0.95)) |>
  t() |>
  data.frame(x = drowning_data$x)  |>
  gather(pct, y, -x)

pfit <- ggplot() +
  geom_point(aes(x, y), data = data.frame(drowning_data), size = 1) +
  geom_line(aes(x, y, linetype = pct), data = mu, color = 'red') +
  scale_linetype_manual(values = c(2,1,2)) +
  labs(title = "5%, 50% and 95% posterior quantiles", y = 'number of drowned', x= "year")
  guides(linetype = "none")

phist_beta <- mcmc_hist(draws2, pars = c('beta'))
phist_ypred <- mcmc_hist(draws2, pars = c('ypred'))

pfit
```



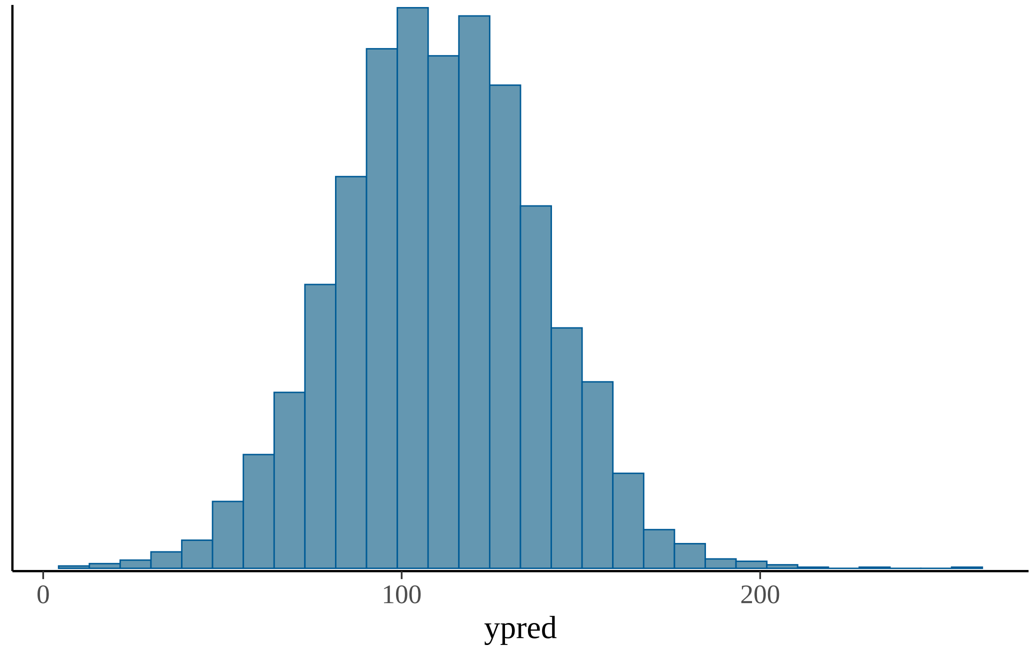
```
phist_beta
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
phist_ypred
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



2. Hierarchical model: factory data with Stan (3 points)

Read in data consisting of 5 quality measurements for 6 machines (units irrelevant):

```r
data("factory")
# some data manipulation to make it suitable
factory_long <- pivot_longer(factory,
                             cols = everything(),
                             names_to = "machine",
                             values_to = "value")
mean_total <- mean(factory_long$value)
sd_total <- sd(factory_long$value)
```

In our factory data we have 5 quality control measurements for 6 different machines. In the separate model each measurement is assumed to come form a different distribution and thus has a different prior mean and standard deviation. In the pooled model all the machines are combined into one group and all their measurements are assumed to come from some shared distribution, so we only define one set of prior mean and sigma. The hierarchical model assumes that the data y_j (quality measurements) are come from a distribution with a common standard deviation but different means for each group. The group means are normally distributed, with mean μ_0 and standard deviation σ_0.

For the given separate Stan model, the default priors don't make sense given the data; we don't know the units or the overall scale of the measurements (faulty products / 1000?), but we know they are all positive, so we should adjust our priors or standarize our data.

Separate model

(a)

The separate model can be summarized by:
observation model $y_{ij} \sim N(\mu_j, \sigma_j)$
mu prior $\mu_j \sim N(93, 10)$
sigma prior $\sigma_j \sim \mathrm{Inv} - \chi^2(10)$

(b)

Separate model stan code:

```
data {
  int<lower=0> N; // number of observations
  int<lower=0> J; // number of groups
  vector[J] y[N];
}

parameters {
  vector[J] mu; // group means
  vector<lower=0>[J] sigma; //group standard deviations
}

model {
  // priors
  for (j in 1:J){
    mu[j] ~ normal(93, 10);
    sigma[j] ~ inv_chi_square(10);
  }

  // likelihood
  for (j in 1:J){
      y[,j] ~ normal(mu[j], sigma[j]);
  }

}
```

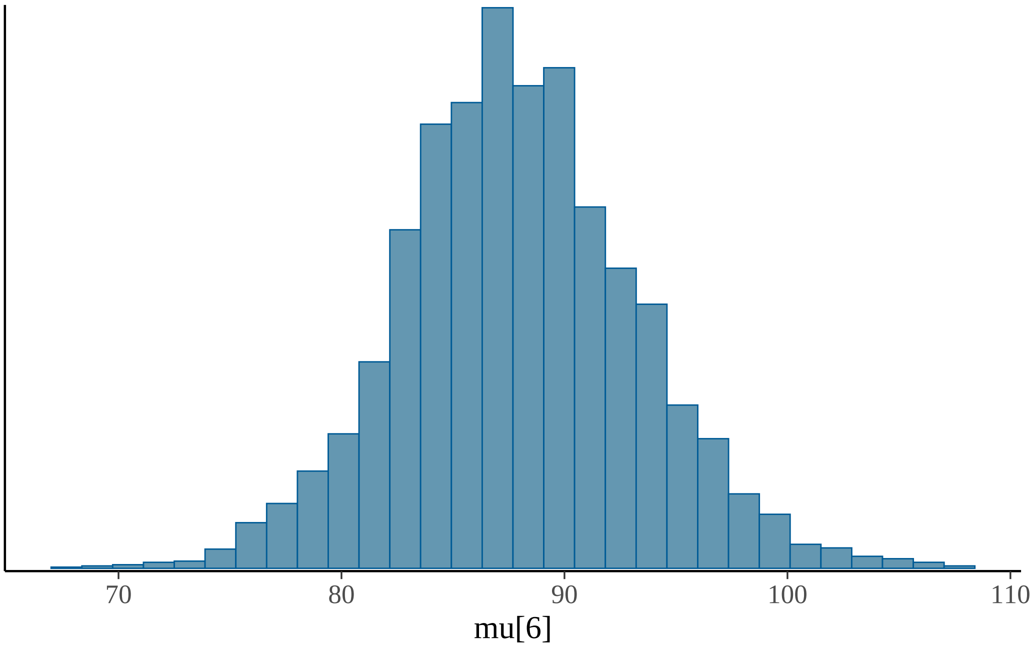
```
generated quantities {
  real ypred;
  // Compute predictive distribution
  // for the sixth machine
  ypred = normal_rng(mu[6], sigma[6]);
}
```

(c)

```
stan_data <- list(N = nrow(factory),
                  J = ncol(factory),
                  y = factory)

model_21 <- cmdstan_model(stan_file = "A7_factory_separate.stan")
fit_21 <- model_21$sample(data = stan_data, refresh = 1000)
draws_21 <- fit_21$draws(format="df")
fit_21$summary()
```

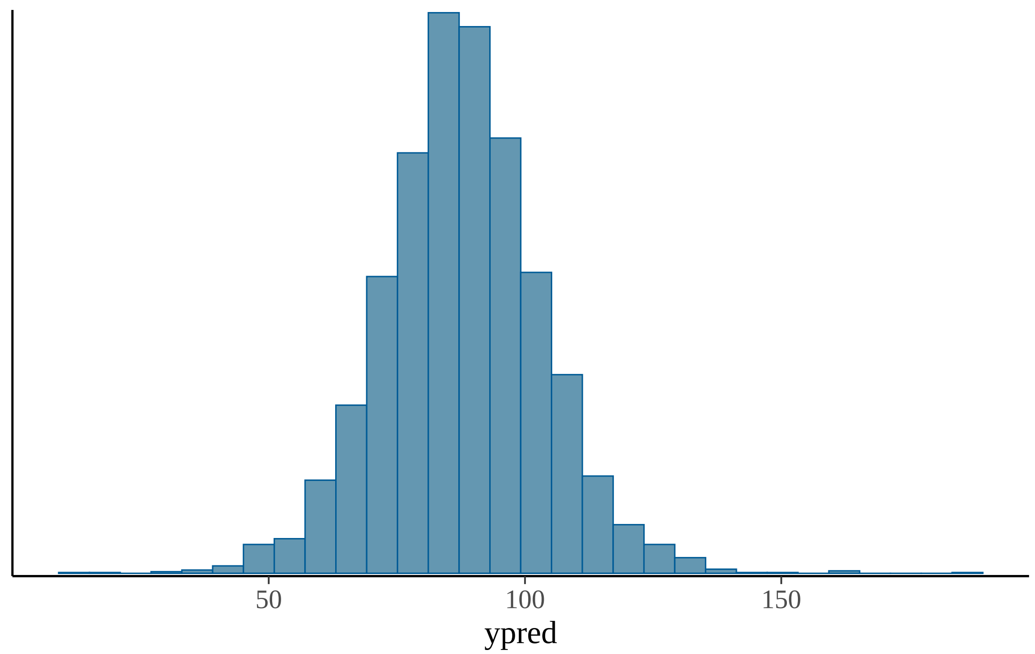
```
mcmc_hist(draws_21, pars = c('mu[6]'))
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
mcmc_hist(draws_21, pars = c('ypred'))
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



(d)

```
stan_data <- list(N = nrow(factory),
                  J = ncol(factory),
                  y = factory)

model_21d <- cmdstan_model(stan_file = "A7_factory_separate_d.stan")
fit_21d <- model_21d$sample(data = stan_data, refresh = 1000, show_messages = FALSE)
draws_21d <- fit_21d$draws(format="df")
```

The 90% credible interval of the posterior expectation for μ_1 is:

```
select(fit_21d$summary(variables=c("mu[1]")), c("q5", "q95"))
```

```
# A tibble: 1 x 2
     q5   q95
  <dbl> <dbl>
1  34.6  64.4
```

Pooled model

(a)

The pooled model can be summarized by:
observation model $y_{ij} \sim N(\mu, \sigma)$
mu prior $\mu \sim N(93, 1)$
sigma prior $\sigma \sim \text{Inv} - \chi^2(10)$

(b)

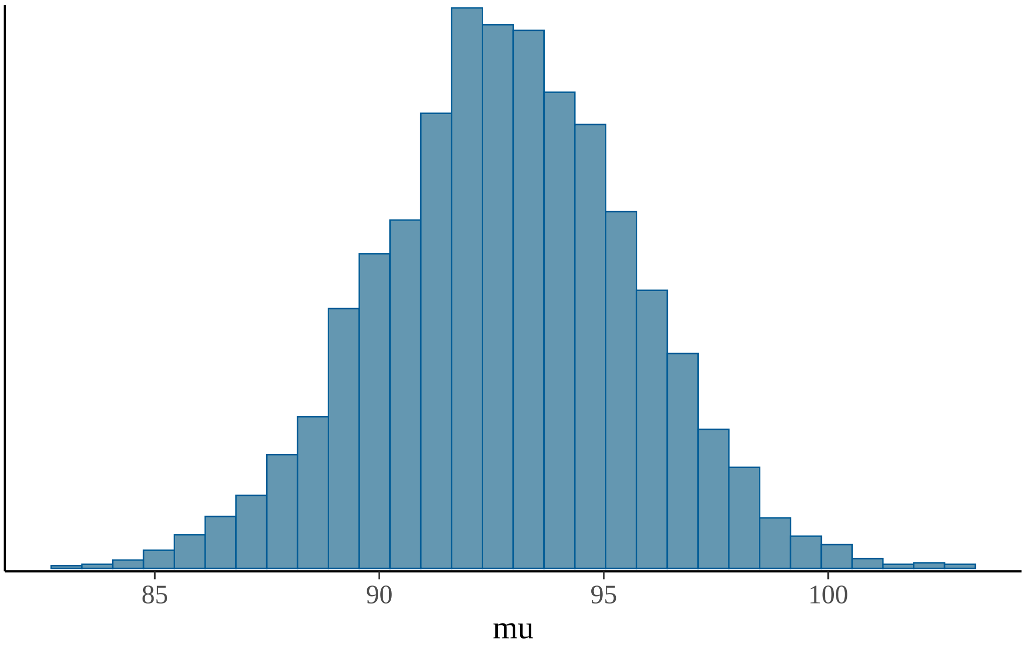
```
data {
  int<lower=0> N; // number of observations
  int<lower=0> J; // number of groups
  vector[N*J] y;
}

parameters {
  real mu; // common mean
  real<lower=0> sigma; // common standard deviation
}

model {
  mu ~ normal(93, 10); // weakly informative prior
  sigma ~ inv_chi_square(10); // weakly informative prior
  y ~ normal(mu, sigma); // observation model / likelihood
}

generated quantities {
  real ypred;
  // Compute predictive distribution
  // for the sixth machine
  ypred = normal_rng(mu, sigma);
}
```

(c)

```
stan_data <- list(N = nrow(factory),
                  J = ncol(factory),
                  y = factory_long$value)

model_22 <- cmdstan_model(stan_file = "A7_factory_pooled.stan")
fit_22 <- model_22$sample(data = stan_data, refresh = 1000, show_messages = FALSE)
draws_22 <- fit_22$draws(format="df")
fit_22$summary()
```

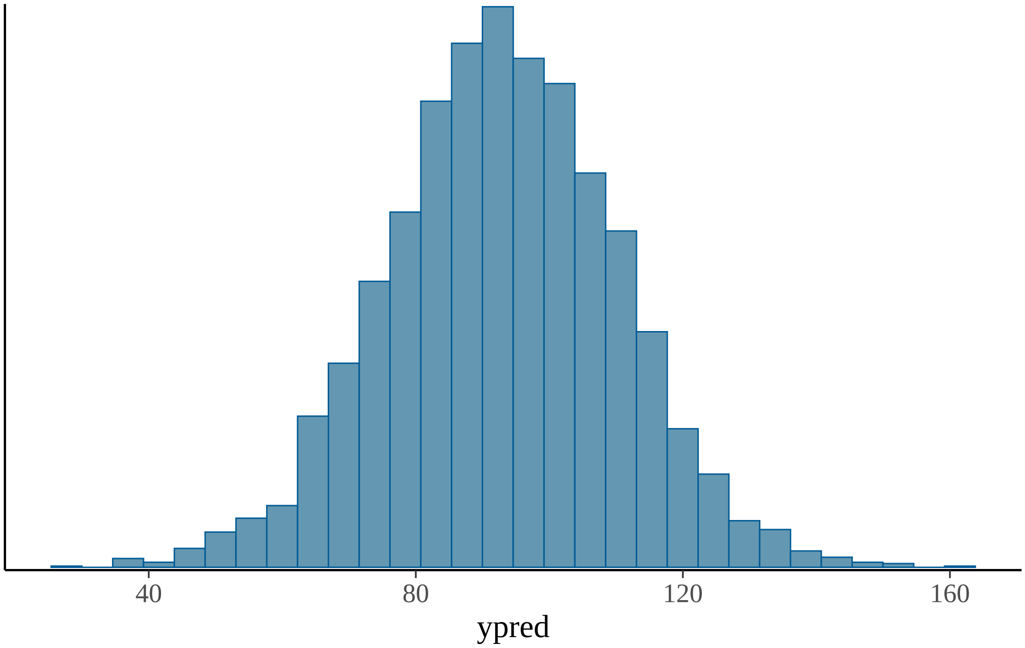
```
mcmc_hist(draws_22, pars = c('mu'))
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
mcmc_hist(draws_22, pars = c('ypred'))
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



(d)

```
stan_data <- list(N = nrow(factory),
                  J = ncol(factory),
                  y = factory_long$value)

model_22d <- cmdstan_model(stan_file = "A7_factory_pooled_d.stan")
fit_22d <- model_22d$sample(data = stan_data, refresh = 1000, show_messages = FALSE)
draws_22d <- fit_22d$draws(format="df")
```

The 90% credible interval of the posterior expectation for μ_1 is:

```
select(fit_22d$summary(variables=c("mu")), c("q5", "q95"))
```

```
# A tibble: 1 x 2
     q5   q95
  <dbl> <dbl>
1  80.2  90.6
```

Hierarchical model

(a)

The hierarchical model can be summarized by:
observation model $y_j \sim N(\mu_j, \sigma)$
mu prior (for population) $\mu_j \sim N(\mu_0, \sigma_0)$
sigma prior (common for population) $\sigma \sim \text{lognormal}(0, 0.5)$
mu prior (for group) $\mu_0 \sim N(93, 10)$
sigma prior (for group) $\sigma_0 \sim N(18, 10)$

(b)

```
data {
  int<lower=0> N; // number of observations
  int<lower=0> J; // number of groups
  int x[N];
  vector[N] y;
}

parameters {
  vector[J] thetaj; // group means
  real mu; // prior mean
  real<lower=0> tau; // prior variance constrained to be positive
  real<lower=0> sigma; // common std constrained to be positive
}

model {
  // mu; // uniform prior as said in BDA3
  // tau; // uniform prior as said in BDA3
  thetaj ~ normal(mu, tau); // population prior with unknown parameters
  // log-normal prior sets normal prior on logarithm of the paremeter,
  // which is useful for positive parameters that shouldn't be very
  // close to 0. BDA3 Chapter 5 uses scaled inverse Chi^2 prior, but
  // as these don't need to be (semi-)conjugate, thinking in terms of
  // log-normal can be easier.
  sigma ~ lognormal(0, .5); // weakly informative prior
  y ~ normal(thetaj[x], sigma); // observation model / likelihood
}

generated quantities {
  real ypred;
  // Compute predictive distribution
  // for the sixth machine
  ypred = normal_rng(thetaj[6], sigma);
}
```

(c)

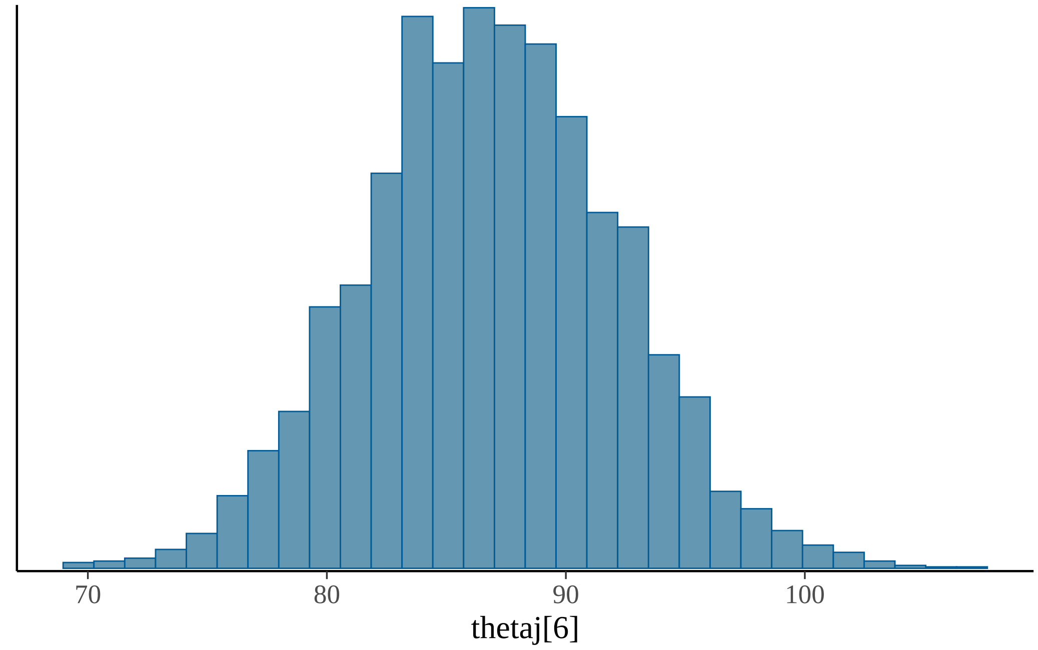
```r
stan_data <- list(N = ncol(factory)*nrow(factory),
                  J = ncol(factory),
                  x = rep(1:ncol(factory), nrow(factory)),
                  y = factory_long$value)

model_23 <- cmdstan_model(stan_file = "A7_factory_hierarchical.stan")
fit_23 <- model_23$sample(data = stan_data, refresh = 1000, show_messages=FALSE)
```

```
Warning: 2 of 4000 (0.0%) transitions ended with a divergence.
See https://mc-stan.org/misc/warnings for details.
```

```r
draws_23 <- fit_23$draws(format="df")
fit_23$summary()
```

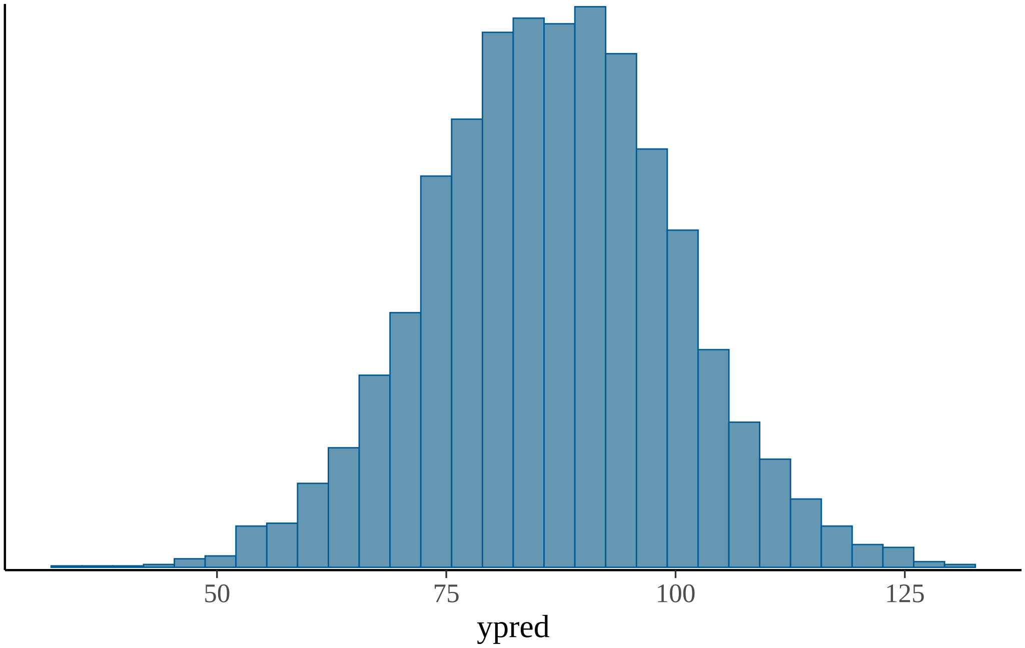
```r
mcmc_hist(draws_23, pars = c('thetaj[6]'))
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
mcmc_hist(draws_23, pars = c('ypred'))
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



(d)

```
stan_data <- list(N = ncol(factory)*nrow(factory),
                  J = ncol(factory),
                  x = rep(1:ncol(factory), nrow(factory)),
                  y = factory_long$value)

model_23d <- cmdstan_model(stan_file = "A7_factory_hierarchical_d.stan")
fit_23d <- model_23d$sample(data = stan_data, refresh = 1000, show_messages = FALSE)
```

```
Warning: 87 of 4000 (2.0%) transitions ended with a divergence.
See https://mc-stan.org/misc/warnings for details.
```

```
draws_23d <- fit_23d$draws(format="df")
```

The 90% credible interval of the posterior expectation for μ_1 (θ_j in the model) is:

```
select(fit_23d$summary(variables="thetaj[1]"), c("q5", "q95"))
```

```
# A tibble: 1 x 2
     q5   q95
  <dbl> <dbl>
1  66.3  84.8
```