Loading Data

```
swim = read.table(url('http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/swim.dat'))
```

Predictive Distribution Construction for informative prior

```
library(MASS)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

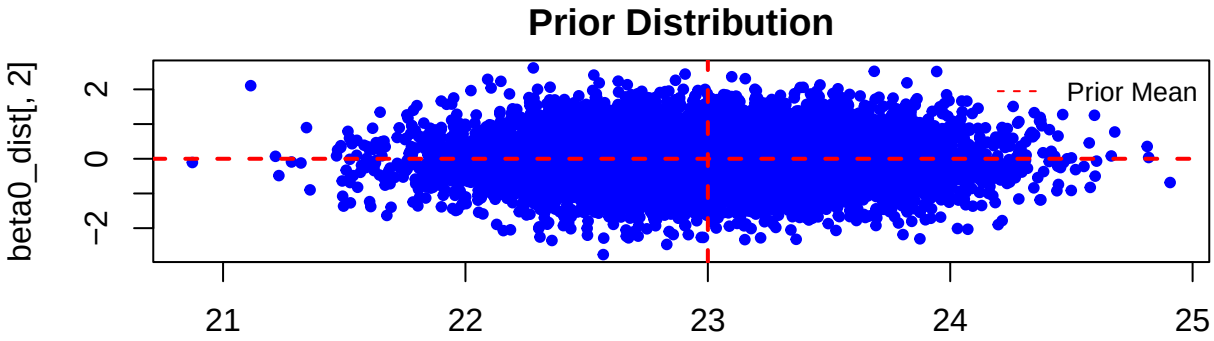
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

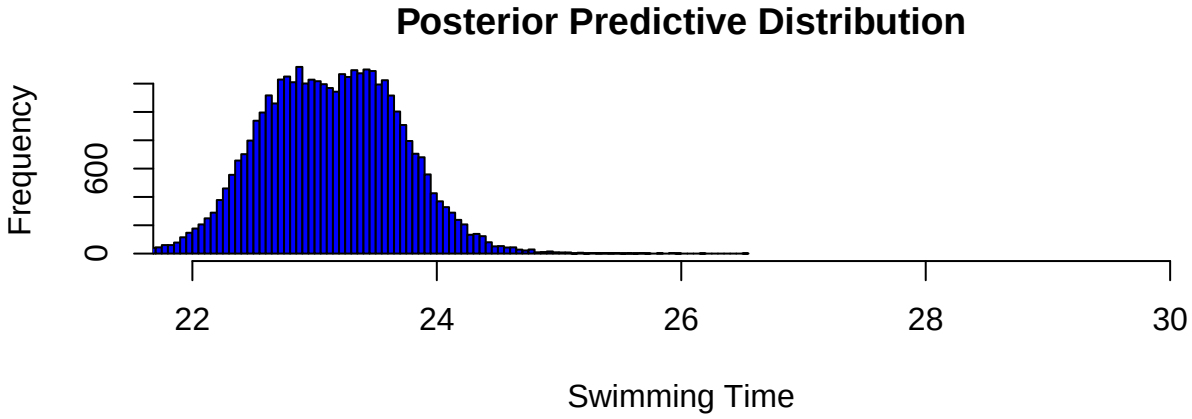
```
S = 10000
X = cbind(rep(1, 6), seq(1, 11, by = 2))
n = dim(X)[1]
p = dim(X)[2]


beta0 = c(23, 0)
sigma0 = rbind(c(0.25, 0), c(0, 0.5))
nu0 = 1
s20 = 0.25

set.seed(8990)

swim_pred = apply(swim, MARGIN = 1, function(y) {

  BETA = matrix(nrow = S, ncol = length(beta0))
  SIGMA = numeric(S)
  beta = c(23, 0)
  s2 = 0.7^2

  # Gibbs algo
  for (s in 1:S) {
    V = solve(solve(sigma0) + (t(X) %*% X) / s2)
    m = V %*% (solve(sigma0) %*% beta0 + (t(X) %*% y) / s2)

    beta = mvrnorm(1, m, V)
    ssr = (t(y) %*% y) - (2 * t(beta) %*% t(X) %*% y) + (t(beta) %*% t(X) %*% X %*% beta)
    s2 = 1 / rgamma(1, (nu0 + n) / 2, (nu0 * s20 + ssr) / 2)

    BETA[s, ] = beta
```

```
    SIGMA[s] = s2
  }

  xpred = c(1, 13)
  YPRED = rnorm(S, BETA %*% xpred, sqrt(SIGMA))

  YPRED
})
```

Plotting the Predictive Distribution and informative prior

```
# Plot posterior predictive and prior distributions
par(mfrow=c(2,1), mar=c(4,4,2,1))

# Prior distributions
beta0_dist = mvrnorm(10000, beta0, sigma0)
plot(beta0_dist[,1], beta0_dist[,2], pch=20, col="blue", main="Prior Distribution")
abline(v=beta0[1], h=beta0[2], lwd=2, col="red", lty=2)
legend("topright", legend="Prior Mean", lty=2, col="red", bty="n", cex=0.8)

# Posterior predictive distributions
hist(swim_pred, main="Posterior Predictive Distribution", xlab="Swimming Time", col="blue", xlim=c(22,3
```





Finding probability of being the best for each of the swimmers in the case of informative prior

```
best_times = apply(swim_pred, MARGIN = 1, FUN = which.min)
table(best_times) / length(best_times)
```

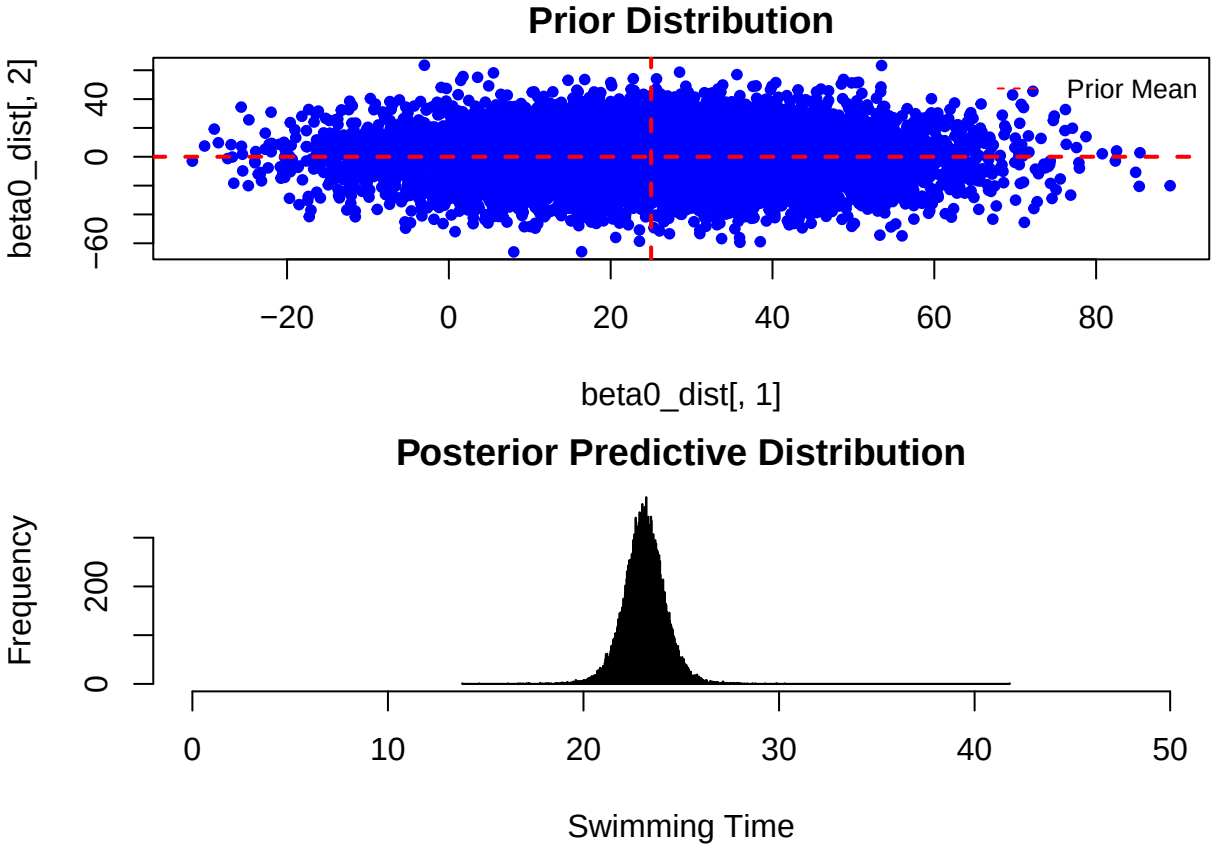
```
## best_times
##      1      2      3      4
## 0.6571 0.0195 0.2977 0.0257
```

Predictive Distribution Construction for flat prior

```
library(MASS)
library(dplyr)

S = 10000
X = cbind(rep(1, 6), seq(1, 11, by = 2))
n = dim(X)[1]
p = dim(X)[2]

beta0 = c(25, 0)
sigma0 = rbind(c(300, 0), c(0, 300))
nu0 = p
s20 = 1
set.seed(8990)

swim_pred = apply(swim, MARGIN = 1, function(y) {

  BETA = matrix(nrow = S, ncol = length(beta0))
  SIGMA = numeric(S)
  beta = c(25,0)
  s2 = 0.7^2

  # Gibbs algo
  for (s in 1:S) {

    V = solve(solve(sigma0) + (t(X) %*% X) / s2)
    m = V %*% (solve(sigma0) %*% beta0 + (t(X) %*% y) / s2)

    beta = mvrnorm(1, m, V)
    ssr = (t(y) %*% y) - (2 * t(beta) %*% t(X) %*% y) + (t(beta) %*% t(X) %*% X %*% beta)
    s2 = 1 / rgamma(1, (nu0 + n) / 2, (nu0 * s20 + ssr) / 2)

    BETA[s, ] = beta
    SIGMA[s] = s2
  }

  xpred = c(1, 13)
  YPRED = rnorm(S, BETA %*% xpred, sqrt(SIGMA))

  YPRED
})
```

Plotting the Predictive Distribution and flat prior

```
# Plot posterior predictive and prior distributions
par(mfrow=c(2,1), mar=c(4,4,2,1))

# Prior distributions
beta0_dist = mvrnorm(10000, beta0, sigma0)
plot(beta0_dist[,1], beta0_dist[,2], pch=20, col="blue", main="Prior Distribution")
abline(v=beta0[1], h=beta0[2], lwd=2, col="red", lty=2)
legend("topright", legend="Prior Mean", lty=2, col="red", bty="n", cex=0.8)

# Posterior predictive distributions
hist(swim_pred, main="Posterior Predictive Distribution", xlab="Swimming Time", col="blue", xlim=c(0,50)
```



Finding probability of being the best for each of the swimmers in the case of flat prior

```
best_times = apply(swim_pred, MARGIN = 1, FUN = which.min)
table(best_times) / length(best_times)
```

```
## best_times
##      1      2      3      4
## 0.4550 0.1052 0.2983 0.1415
```

Bayesian Linear Regression of swim time vs week for all swimmers together with no distinction between them

```
swim <- c(23.1, 23.2, 22.9, 22.9, 22.8, 22.7,
          23.2, 23.1, 23.4, 23.5, 23.5, 23.4,
          22.7, 22.6, 22.8, 22.8, 22.9, 22.8,
          23.7, 23.6, 23.7, 23.5, 23.5, 23.4)
weeks <- rep(c(1,3,5,7, 9, 11), each = 4)
df <- data.frame(weeks, swim)

library(rstan)
```

```
## Loading required package: StanHeaders
```

```
## Loading required package: ggplot2
```

```
## rstan (Version 2.21.7, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
# Define the Bayesian linear regression model
model_code <- "
data {
  int<lower=0> N;           // number of data points
  vector[N] y;              // response variable
  vector[N] x;              // explanatory variable
}
parameters {
  real alpha;               // intercept
  real beta;                // slope
  real<lower=0> sigma;      // residual standard deviation
}
model {
  alpha ~ normal(23, 0.25); // prior for intercept
  beta ~ normal(0, 10);     // prior for slope
  sigma ~ cauchy(0, 5);     // prior for residual standard deviation
  y ~ normal(alpha + beta * x, sigma); // likelihood
}
"

# Prepare the data
N <- nrow(df)
y <- df$swim
x <- df$weeks

# Run the model using Stan
model_data <- list(N = N, y = y, x = x)
model_fit <- stan(model_code = model_code, data = model_data, chains = 4, iter = 2000)
```

```
## Trying to compile a simple C file
```

```
## Running /usr/local/R-4.2.1/lib64/R/bin/R CMD SHLIB foo.c
## gcc -I"/usr/local/R-4.2.1/lib64/R/include" -DNDEBUG   -I"/usr/local/R-4.2.1/lib64/R/library/Rcpp/incl
```

```
## In file included from /usr/local/R-4.2.1/lib64/R/library/RcppEigen/include/Eigen/Core:88:0,
##                  from /usr/local/R-4.2.1/lib64/R/library/RcppEigen/include/Eigen/Dense:1,
##                  from /usr/local/R-4.2.1/lib64/R/library/StanHeaders/include/stan/math/prim/mat/fun/
##                  from <command-line>:0:
## /usr/local/R-4.2.1/lib64/R/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:628:1: error: unkn
##  namespace Eigen {
##  ^~~~~~~~~
## /usr/local/R-4.2.1/lib64/R/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:628:17: error: exp
##  namespace Eigen {
##                  ^
## In file included from /usr/local/R-4.2.1/lib64/R/library/RcppEigen/include/Eigen/Dense:1:0,
##                  from /usr/local/R-4.2.1/lib64/R/library/StanHeaders/include/stan/math/prim/mat/fun/
##                  from <command-line>:0:
## /usr/local/R-4.2.1/lib64/R/library/RcppEigen/include/Eigen/Core:96:10: fatal error: complex: No such
##  #include <complex>
##           ^~~~~~~~~
## compilation terminated.
## make: *** [/usr/local/R-4.2.1/lib64/R/etc/Makeconf:168: foo.o] Error 1
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.04 seconds (Warm-up)
## Chain 1:                0.04 seconds (Sampling)
## Chain 1:                0.08 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
```

```
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.04 seconds (Warm-up)
## Chain 2:                0.05 seconds (Sampling)
## Chain 2:                0.09 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.05 seconds (Warm-up)
## Chain 3:                0.04 seconds (Sampling)
## Chain 3:                0.09 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
```

```
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.05 seconds (Warm-up)
## Chain 4:                0.03 seconds (Sampling)
## Chain 4:                0.08 seconds (Total)
## Chain 4:
```

```
# Print the summary of the model results
print(summary(model_fit))
```

```
## $summary
##              mean      se_mean         sd         2.5%         25%         50%
## alpha 22.94361087 0.0031863620 0.12612355 22.699182321 22.86279506 22.94251492
## beta   0.03575939 0.0004652442 0.01915312 -0.002134592  0.02327122  0.03614761
## sigma  0.35574650 0.0011968855 0.05601454  0.267581676  0.31556660  0.34777343
## lp__  12.54859269 0.0329876291 1.27968226  9.156544413 11.95608151 12.88071967
##               75%      97.5%    n_eff     Rhat
## alpha 23.02693499 23.1958918 1566.759 1.002056
## beta   0.04841662  0.0737232 1694.795 1.003140
## sigma  0.38764297  0.4849764 2190.263 1.000872
## lp__  13.47474779 14.0102075 1504.881 1.002798
##
## $c_summary
## , , chains = chain:1
##
##          stats
## parameter        mean         sd         2.5%         25%         50%
##     alpha 22.94435636 0.12279033 22.687110830 22.86653277 22.94119221
##     beta   0.03568446 0.01865545 -0.002465876  0.02413069  0.03614186
##     sigma  0.35440241 0.05436055  0.268858129  0.31485623  0.34742729
##     lp__  12.63661979 1.20600231  9.557777955 12.04334467 12.95160318
##          stats
## parameter        75%       97.5%
##     alpha 23.0215339 23.18571680
##     beta   0.0473789  0.07320376
##     sigma  0.3836003  0.48033694
##     lp__  13.5542509 14.03570170
##
## , , chains = chain:2
##
##          stats
## parameter        mean         sd         2.5%         25%         50%
##     alpha 22.94375134 0.13131828 22.700844090 22.86002736 22.93894164
##     beta   0.03544921 0.01974608 -0.003338979  0.02203928  0.03658973
##     sigma  0.35587777 0.05570627  0.268072445  0.31827824  0.34797676
##     lp__  12.55534937 1.37622413  8.733943368 11.99466574 12.92288093
##          stats
## parameter         75%       97.5%
##     alpha 23.03513478 23.20294032
##     beta   0.04873357  0.07189764
##     sigma  0.38446481  0.48580533
```

```
##     lp__  13.50980003 14.01580744
##
## , , chains = chain:3
##
##          stats
## parameter       mean         sd         2.5%         25%         50%
##     alpha 22.9398361 0.11760709 2.271538e+01 22.85902658 22.94237290
##     beta   0.0365755 0.01783852 8.396437e-04  0.02500285  0.03658332
##     sigma  0.3528455 0.05761016 2.653942e-01  0.31179424  0.34064060
##     lp__  12.5807975 1.23106126 9.361538e+00 12.03123293 12.88531669
##          stats
## parameter         75%       97.5%
##     alpha 23.02049627 23.17556987
##     beta   0.04850067  0.06964576
##     sigma  0.38682182  0.48196203
##     lp__  13.44541870 13.99811723
##
## , , chains = chain:4
##
##          stats
## parameter        mean         sd         2.5%         25%         50%
##     alpha 22.94649973 0.13228680 22.685100894 22.86451830 22.95220522
##     beta   0.03532841 0.02028225 -0.002135546  0.02149578  0.03488018
##     sigma  0.35986031 0.05617435  0.271204623  0.31940739  0.35491364
##     lp__  12.42160413 1.29086621  9.022614604 11.81087172 12.75353686
##          stats
## parameter         75%       97.5%
##     alpha 23.03360935 23.20658449
##     beta   0.04908718  0.07741404
##     sigma  0.39482028  0.48636055
##     lp__  13.37584510 13.96516252
```

Bayesian Linear Regression for Swimmer 1

```
swim <- c(23.1, 23.2, 22.9, 22.9, 22.8, 22.7)
weeks <- c(1,3,5,7, 9, 11)
df <- data.frame(weeks, swim)

library(rstan)

# Define the Bayesian linear regression model
model_code <- "
data {
  int<lower=0> N;           // number of data points
  vector[N] y;              // response variable
  vector[N] x;              // explanatory variable
}
parameters {
  real alpha;               // intercept
  real beta;                // slope
  real<lower=0> sigma;      // residual standard deviation
}
model {
```

```
  alpha ~ normal(23, 0.25); // prior for intercept
  beta ~ normal(0, 10);     // prior for slope
  sigma ~ cauchy(0, 5);     // prior for residual standard deviation
  y ~ normal(alpha + beta * x, sigma); // likelihood
}
"

# Prepare the data
N <- nrow(df)
y <- df$swim
x <- df$weeks

# Run the model using Stan
model_data <- list(N = N, y = y, x = x)
model_fit <- stan(model_code = model_code, data = model_data, chains = 4, iter = 2000)
```

```
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.05 seconds (Warm-up)
## Chain 1:                0.04 seconds (Sampling)
## Chain 1:                0.09 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
```

```
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.04 seconds (Warm-up)
## Chain 2:                0.03 seconds (Sampling)
## Chain 2:                0.07 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.04 seconds (Warm-up)
## Chain 3:                0.03 seconds (Sampling)
## Chain 3:                0.07 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

```
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.04 seconds (Warm-up)
## Chain 4:                0.04 seconds (Sampling)
## Chain 4:                0.08 seconds (Total)
## Chain 4:
```

```
# Print the summary of the model results
print(summary(model_fit))
```

```
## $summary
##              mean      se_mean         sd        2.5%         25%         50%
## alpha 23.17283747 0.0030375873 0.10042234 22.94267628 23.12018691 23.18176044
## beta  -0.04141811 0.0004501125 0.01488823 -0.06835037 -0.05024958 -0.04252708
## sigma  0.12590303 0.0023218826 0.07134322  0.05361314  0.08154022  0.10614916
## lp__   8.13914774 0.0583964820 1.62252061  4.18740205  7.35532448  8.51990115
##              75%        97.5%     n_eff     Rhat
## alpha 23.2318463 23.351854852 1092.9572 1.005349
## beta  -0.0335550 -0.008175123 1094.0673 1.005921
## sigma  0.1479038  0.306252714  944.1149 1.004223
## lp__   9.3490546 10.072401849  771.9818 1.003918
##
## $c_summary
## , , chains = chain:1
##
##          stats
## parameter        mean         sd        2.5%         25%         50%
##     alpha 23.17057575 0.09315535 22.95934992 23.11687547 23.17780308
##     beta  -0.04135181 0.01375000 -0.06676127 -0.05026524 -0.04197662
##     sigma  0.13272889 0.08418298  0.05350761  0.08125169  0.11095215
##     lp__   7.98049058 1.69971863  3.81225705  7.16265566  8.34230715
##          stats
## parameter         75%       97.5%
##     alpha 23.23017620 23.33834402
##     beta  -0.03350886 -0.01118947
##     sigma  0.15888260  0.32326230
##     lp__   9.24148530 10.05649903
##
## , , chains = chain:2
##
##          stats
## parameter       mean         sd        2.5%         25%         50%         75%
##     alpha 23.1736458 0.09754493 22.93691001 23.12034180 23.18917782 23.23141611
##     beta  -0.0416325 0.01443666 -0.06808302 -0.05026062 -0.04337766 -0.03404188
##     sigma  0.1216422 0.05774941  0.05431328  0.08248204  0.10657169  0.14163235
##     lp__   8.3206222 1.42492803  4.76761515  7.62592641  8.63207997  9.41810798
##          stats
## parameter        97.5%
##     alpha 23.338761126
##     beta  -0.007596961
##     sigma  0.268442873
##     lp__  10.074347970
##
```

```
## , , chains = chain:3
##
##          stats
## parameter        mean         sd        2.5%         25%         50%
##     alpha 23.16454563 0.10909195 22.89534892 23.11549030 23.17739807
##     beta  -0.04004009 0.01583370 -0.06482816 -0.04918019 -0.04177068
##     sigma  0.12650480 0.07542115  0.05312851  0.08078552  0.10426988
##     lp__   8.10149374 1.73878801  3.53414884  7.28516933  8.52490101
##          stats
## parameter         75%       97.5%
##     alpha 23.22652066 23.33561583
##     beta  -0.03225621 -0.00485823
##     sigma  0.14311470  0.33169144
##     lp__   9.37946245 10.08863243
##
## , , chains = chain:4
##
##          stats
## parameter        mean         sd        2.5%         25%         50%
##     alpha 23.18258267 0.10052474 22.96309771 23.13208639 23.18369446
##     beta  -0.04264803 0.01535098 -0.07402150 -0.05130684 -0.04298333
##     sigma  0.12273627 0.06472536  0.05432916  0.08128647  0.10464959
##     lp__   8.15398442 1.59234050  4.28399783  7.34862085  8.50006638
##          stats
## parameter         75%        97.5%
##     alpha 23.24324417 23.375961846
##     beta  -0.03459137 -0.009986522
##     sigma  0.14306458  0.293915150
##     lp__   9.35801403 10.067375966
```

Bayesian Linear Regression for Swimmer 2

```
swim <- c(23.2, 23.1, 23.4, 23.5, 23.5, 23.4)
weeks <- c(1,3,5,7, 9, 11)
df <- data.frame(weeks, swim)

library(rstan)

# Define the Bayesian linear regression model
model_code <- "
data {
  int<lower=0> N;           // number of data points
  vector[N] y;              // response variable
  vector[N] x;              // explanatory variable
}
parameters {
  real alpha;               // intercept
  real beta;                // slope
  real<lower=0> sigma;      // residual standard deviation
}
model {
  alpha ~ normal(23, 0.25); // prior for intercept
  beta ~ normal(0, 10);     // prior for slope
```

```
  sigma ~ cauchy(0, 5);     // prior for residual standard deviation
  y ~ normal(alpha + beta * x, sigma); // likelihood
}
"

# Prepare the data
N <- nrow(df)
y <- df$swim
x <- df$weeks

# Run the model using Stan
model_data <- list(N = N, y = y, x = x)
model_fit <- stan(model_code = model_code, data = model_data, chains = 4, iter = 2000)
```

```
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.03 seconds (Warm-up)
## Chain 1:                0.04 seconds (Sampling)
## Chain 1:                0.07 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
```

```
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.04 seconds (Warm-up)
## Chain 2:                0.04 seconds (Sampling)
## Chain 2:                0.08 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.04 seconds (Warm-up)
## Chain 3:                0.02 seconds (Sampling)
## Chain 3:                0.06 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
```

```
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.04 seconds (Warm-up)
## Chain 4:                0.04 seconds (Sampling)
## Chain 4:                0.08 seconds (Total)
## Chain 4:
```

```
# Print the summary of the model results
print(summary(model_fit))
```

```
## $summary
##             mean      se_mean         sd         2.5%         25%         50%
## alpha 23.1133949 0.0038678359 0.12503593 22.846296504 23.04041934 23.11766079
## beta   0.0379997 0.0006116161 0.01989750  0.000649369  0.02608656  0.03720077
## sigma  0.1747145 0.0029572530 0.09603457  0.080376387  0.11559972  0.14882451
## lp__   6.4784072 0.0572562460 1.54715791  2.411873278  5.77796030  6.85816968
##              75%       97.5%     n_eff     Rhat
## alpha 23.1918867 23.34774223 1045.0416 1.004697
## beta   0.0490406  0.08060593 1058.3746 1.004929
## sigma  0.2020530  0.42426342 1054.5769 1.003440
## lp__   7.6143979  8.25945688  730.1693 1.006613
##
## $c_summary
## , , chains = chain:1
##
##          stats
## parameter        mean         sd        2.5%         25%         50%
##     alpha 23.11319995 0.12850500 22.83296713 23.04678859 23.11629864
##     beta   0.03850953 0.02039189 -0.00233836  0.02683019  0.03845021
##     sigma  0.18152798 0.10343244  0.07958115  0.11316662  0.15181490
##     lp__   6.36599385 1.68638612  2.09428720  5.62879629  6.84599485
##          stats
## parameter         75%       97.5%
##     alpha 23.18537454 23.35774550
##     beta   0.04919053  0.08122806
##     sigma  0.21163197  0.48281695
##     lp__   7.59336033  8.26720632
##
## , , chains = chain:2
##
##          stats
## parameter        mean         sd         2.5%         25%         50%
##     alpha 23.10243723 0.13422706 2.281023e+01 23.02662130 23.11061669
##     beta   0.03981586 0.02089997 3.420913e-04  0.02760734  0.03918982
##     sigma  0.17857179 0.09762565 8.431047e-02  0.12007431  0.14880283
##     lp__   6.37767766 1.61048543 2.115241e+00  5.62295708  6.79432379
##          stats
## parameter         75%       97.5%
##     alpha 23.19590349 23.33351961
##     beta   0.05176488  0.08144317
##     sigma  0.20366828  0.43715703
##     lp__   7.60142268  8.24243428
##
## , , chains = chain:3
##
```

```
##          stats
## parameter        mean         sd         2.5%         25%         50%
##     alpha 23.12661427 0.11523478 22.895532254 23.05994970 23.12358708
##     beta   0.03582035 0.01944767 -0.001936812  0.02460543  0.03558375
##     sigma  0.17274413 0.10611258  0.080469042  0.11445463  0.14644025
##     lp__   6.53612081 1.55288339  2.439523277  5.86285044  6.95720605
##          stats
## parameter         75%       97.5%
##     alpha 23.19419740 23.35343316
##     beta   0.04616493  0.07631422
##     sigma  0.19211306  0.41365190
##     lp__   7.64105227  8.25945275
##
## , , chains = chain:4
##
##          stats
## parameter        mean         sd         2.5%         25%        50%
##     alpha 23.11132809 0.12030778 22.868412324 23.03647297 23.1164767
##     beta   0.03785306 0.01859151  0.004032104  0.02611745  0.0364718
##     sigma  0.16601400 0.07272079  0.075898475  0.11492436  0.1471506
##     lp__   6.63383646 1.29716911  3.416957381  6.04064447  6.8846522
##          stats
## parameter         75%       97.5%
##     alpha 23.19367459 23.32610805
##     beta   0.04831353  0.07706456
##     sigma  0.19679297  0.36291827
##     lp__   7.61045455  8.26416472
```

Bayesian Linear Regression for Swimmer 3

```
swim <- c(22.7, 22.6, 22.8, 22.8, 22.9, 22.8)
weeks <- c(1,3,5,7, 9, 11)
df <- data.frame(weeks, swim)

library(rstan)

# Define the Bayesian linear regression model
model_code <- "
data {
  int<lower=0> N;           // number of data points
  vector[N] y;              // response variable
  vector[N] x;              // explanatory variable
}
parameters {
  real alpha;               // intercept
  real beta;                // slope
  real<lower=0> sigma;      // residual standard deviation
}
model {
  alpha ~ normal(23, 0.25); // prior for intercept
  beta ~ normal(0, 10);     // prior for slope
  sigma ~ cauchy(0, 5);     // prior for residual standard deviation
  y ~ normal(alpha + beta * x, sigma); // likelihood
```

```
}
"

# Prepare the data
N <- nrow(df)
y <- df$swim
x <- df$weeks

# Run the model using Stan
model_data <- list(N = N, y = y, x = x)
model_fit <- stan(model_code = model_code, data = model_data, chains = 4, iter = 2000)
```

```
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.05 seconds (Warm-up)
## Chain 1:                0.03 seconds (Sampling)
## Chain 1:                0.08 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

```
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.05 seconds (Warm-up)
## Chain 2:                0.03 seconds (Sampling)
## Chain 2:                0.08 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.05 seconds (Warm-up)
## Chain 3:                0.04 seconds (Sampling)
## Chain 3:                0.09 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
```

```
## Chain 4:  Elapsed Time: 0.04 seconds (Warm-up)
## Chain 4:                0.04 seconds (Sampling)
## Chain 4:                0.08 seconds (Total)
## Chain 4:
```

```
## Warning: There were 6 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
# Print the summary of the model results
print(summary(model_fit))
```

```
## $summary
##             mean      se_mean         sd        2.5%          25%         50%
## alpha 22.7067500 0.0042205357 0.11092123 22.53267703 22.637990824 22.68763954
## beta   0.0122495 0.0006501766 0.01636424 -0.02795323  0.005046723  0.01481201
## sigma  0.1262053 0.0032362430 0.07957306  0.04983239  0.078330345  0.10412729
## lp__   7.6430669 0.0682643126 1.69299647  3.28693148  6.833782640  8.06065361
##               75%       97.5%    n_eff     Rhat
## alpha 22.75456792 22.98810878 690.7075 1.000023
## beta   0.02213037  0.03799356 633.4745 1.001760
## sigma  0.14462485  0.35602714 604.5746 1.004236
## lp__   8.91775166  9.60433938 615.0702 1.005130
##
## $c_summary
## , , chains = chain:1
##
##          stats
## parameter        mean         sd        2.5%          25%         50%
##     alpha 22.70902488 0.11476865 22.53002500 22.636821906 22.68791039
##     beta   0.01176863 0.01690941 -0.03239489  0.005030563  0.01474489
##     sigma  0.12769801 0.07738421  0.05029329  0.079921369  0.10624928
##     lp__   7.60616454 1.63223433  3.50536474  6.755677496  8.05973710
##          stats
## parameter         75%      97.5%
##     alpha 22.75433913 23.0293781
##     beta   0.02185674  0.0377880
##     sigma  0.14805306  0.3453011
##     lp__   8.84646689  9.6156195
##
## , , chains = chain:2
##
##          stats
## parameter        mean         sd        2.5%          25%         50%
##     alpha 22.70408246 0.10811690 22.51575672 22.634879073 22.69175574
##     beta   0.01306003 0.01594244 -0.02539449  0.005168319  0.01488889
##     sigma  0.12900596 0.08370007  0.04873505  0.076503935  0.10439584
##     lp__   7.53915483 1.74309727  3.18937240  6.723395986  7.92316181
##          stats
## parameter        75%       97.5%
##     alpha 22.7565735 22.95830982
##     beta   0.0222561  0.04275179
##     sigma  0.1512901  0.36661283
##     lp__   8.8962778  9.58137680
```

```
##
## , , chains = chain:3
##
##          stats
## parameter        mean         sd        2.5%          25%         50%
##     alpha 22.70677588 0.10487127 22.55415289 22.641546709 22.68464127
##     beta   0.01210283 0.01563330 -0.02417021  0.005392168  0.01466471
##     sigma  0.11743012 0.06606229  0.04868270  0.076212705  0.10052048
##     lp__   7.83692702 1.48127347  4.07892585  7.075279695  8.16341752
##          stats
## parameter         75%       97.5%
##     alpha 22.75341341 22.94748830
##     beta   0.02193322  0.03420268
##     sigma  0.13568033  0.29098511
##     lp__   8.96950571  9.61144631
##
## , , chains = chain:4
##
##          stats
## parameter        mean         sd        2.5%          25%         50%
##     alpha 22.70711672 0.11566976 22.52399704 22.638724739 22.68620867
##     beta   0.01206652 0.01692702 -0.03299777  0.004709917  0.01504475
##     sigma  0.13068696 0.08877495  0.05455943  0.081315940  0.10502521
##     lp__   7.59002134 1.87728521  2.43887556  6.851067100  8.09884754
##          stats
## parameter         75%       97.5%
##     alpha 22.75153129 23.00115967
##     beta   0.02224334  0.03847601
##     sigma  0.14557748  0.37738052
##     lp__   8.93536079  9.61421215
```

Bayesian Linear Regression for Swimmer 4

```
swim <- c(23.7, 23.6, 23.7, 23.5, 23.5, 23.4)
weeks <- c(1,3,5,7, 9, 11)
df <- data.frame(weeks, swim)

library(rstan)

# Define the Bayesian linear regression model
model_code <- "
data {
  int<lower=0> N;           // number of data points
  vector[N] y;              // response variable
  vector[N] x;              // explanatory variable
}
parameters {
  real alpha;               // intercept
  real beta;                // slope
  real<lower=0> sigma;      // residual standard deviation
}
model {
  alpha ~ normal(23, 0.25); // prior for intercept
```

```
  beta ~ normal(0, 10);     // prior for slope
  sigma ~ cauchy(0, 5);     // prior for residual standard deviation
  y ~ normal(alpha + beta * x, sigma); // likelihood
}
"

# Prepare the data
N <- nrow(df)
y <- df$swim
x <- df$weeks

# Run the model using Stan
model_data <- list(N = N, y = y, x = x)
model_fit <- stan(model_code = model_code, data = model_data, chains = 4, iter = 2000)
```

```
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.04 seconds (Warm-up)
## Chain 1:                0.05 seconds (Sampling)
## Chain 1:                0.09 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
```

```
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.05 seconds (Warm-up)
## Chain 2:                0.02 seconds (Sampling)
## Chain 2:                0.07 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.04 seconds (Warm-up)
## Chain 3:                0.05 seconds (Sampling)
## Chain 3:                0.09 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '4d98b59ff89bdce573b5c4998f4fae37' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
```

```
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.04 seconds (Warm-up)
## Chain 4:                0.04 seconds (Sampling)
## Chain 4:                0.08 seconds (Total)
## Chain 4:
```

```
## Warning: There were 107 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quanti
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess
```

```
# Print the summary of the model results
print(summary(model_fit))
```

```
## $summary
##              mean     se_mean         sd        2.5%         25%         50%
## alpha 23.60365842 0.011044560 0.18312153 23.07795461 23.55174285 23.65862869
## beta  -0.01162793 0.001498115 0.02570584 -0.04112579 -0.02665913 -0.01885569
## sigma  0.14717769 0.009108608 0.13556451  0.04757681  0.07261293  0.10104207
## lp__   5.14804137 0.143688100 1.89747446  0.70103098  4.07456864  5.59897835
##                75%      97.5%    n_eff     Rhat
## alpha 23.714180893 23.8083875 274.9046 1.022883
## beta  -0.005214084  0.0620767 294.4239 1.019724
## sigma  0.168801317  0.5213543 221.5073 1.033354
## lp__   6.641600688  7.4511645 174.3855 1.035484
##
## $c_summary
## , , chains = chain:1
##
##          stats
## parameter         mean        sd        2.5%         25%         50%
##     alpha 23.559622293 0.2225609 22.95412241 23.49286017 23.62555560
##     beta  -0.005852079 0.0323803 -0.04125491 -0.02554156 -0.01517232
##     sigma  0.184710718 0.1892855  0.04456444  0.07584260  0.12767640
##     lp__   4.603747439 2.1799523 -0.64919090  3.28800608  5.06129640
##          stats
## parameter          75%       97.5%
##     alpha 23.702354171 23.80577547
##     beta   0.002441506  0.08649199
##     sigma  0.212924667  0.68920860
##     lp__   6.318064794  7.37338323
##
## , , chains = chain:2
##
##          stats
## parameter       mean         sd        2.5%         25%         50%
```

```
##     alpha 23.6153062 0.17279464 23.10634422 23.57961499 23.67095855
##     beta  -0.0133415 0.02428381 -0.04150271 -0.02707157 -0.02048381
##     sigma  0.1402738 0.11486909  0.05078297  0.07384890  0.09879361
##     lp__   5.3323860 1.78216883  0.93300398  4.41810560  5.76885247
##          stats
## parameter          75%       97.5%
##     alpha 23.713548828 23.81149273
##     beta  -0.008306804  0.06375764
##     sigma  0.156069726  0.48825942
##     lp__   6.757557714  7.46489857
##
## , , chains = chain:3
##
##          stats
## parameter        mean         sd        2.5%         25%         50%
##     alpha 23.60727937 0.17576764 23.09181731 23.54674277 23.66047489
##     beta  -0.01207787 0.02369086 -0.03980740 -0.02627855 -0.01881437
##     sigma  0.14178421 0.11749546  0.04750226  0.07206229  0.09975222
##     lp__   5.19323817 1.84446415  0.82568744  4.13030357  5.68478829
##          stats
## parameter          75%       97.5%
##     alpha 23.714498239 23.80418817
##     beta  -0.005427987  0.05438834
##     sigma  0.166261266  0.47745458
##     lp__   6.641600688  7.47732354
##
## , , chains = chain:4
##
##          stats
## parameter        mean         sd        2.5%         25%         50%
##     alpha 23.63242581 0.14501946 23.19028553 23.58721314 23.66759243
##     beta  -0.01524028 0.01990236 -0.04123471 -0.02713991 -0.01996218
##     sigma  0.12194202 0.09280412  0.04815052  0.06826994  0.09248438
##     lp__   5.46279384 1.62797461  1.65035231  4.52698078  5.83717350
##          stats
## parameter          75%       97.5%
##     alpha 23.723959361 23.80649269
##     beta  -0.007537181  0.04099418
##     sigma  0.137535574  0.39674076
##     lp__   6.757539291  7.46533397
```