2020 SISG Bayesian Statistics for Genetics R Notes: Multinomial Sampling

Jon Wakefield
Departments of Statistics and Biostatistics, University of Washington

2020-07-21

Hardy-Weinberg via Fisher's exact test

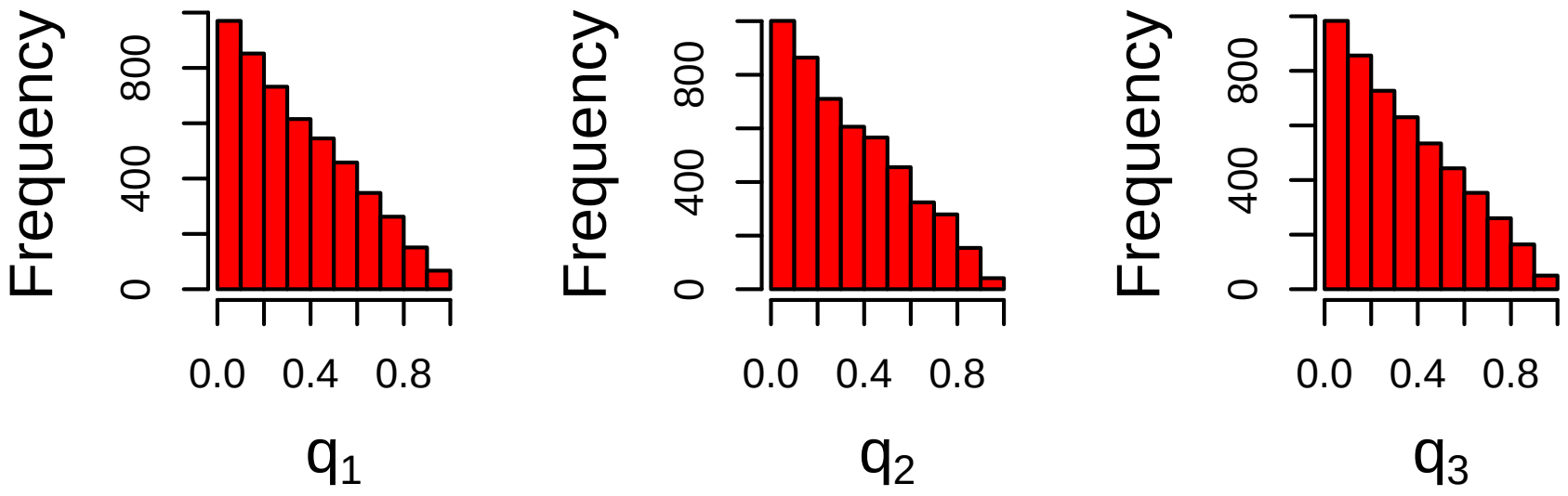
```
library(hwde)
n1 <- 88
n2 <- 10
n3 <- 2
exact <- hwexact(n1, n2, n3)
exact
## [1] 0.06544427
```

We obtain a p-value of 0.07

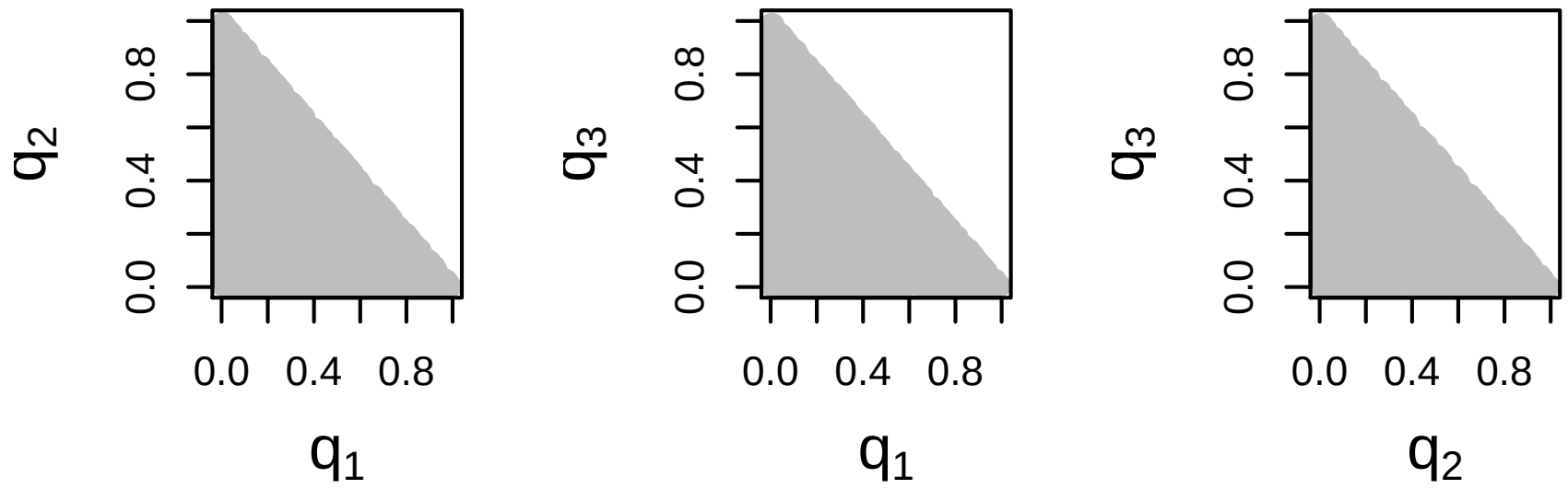
Displaying samples from a dirichlet(1,1,1)

```
library(VGAM)  # To access the rdiric function
nsim <- 5000
q <- rdiric(nsim, c(1, 1, 1))
# Univariate marginal representations
par(mfrow = c(1, 3))
hist(q[, 1], xlab = expression(q[1]), main = "", cex.lab = 1.5,
    xlim = c(0, 1), col = "red")
hist(q[, 2], xlab = expression(q[2]), main = "", cex.lab = 1.5,
    xlim = c(0, 1), col = "red")
hist(q[, 3], xlab = expression(q[3]), main = "", cex.lab = 1.5,
    xlim = c(0, 1), col = "red")
# Bivariate representations
plot(q[, 1], q[, 2], xlim = c(0, 1), ylim = c(0, 1),
    xlab = expression(q[1]), ylab = expression(q[2]),
    cex.lab = 1.5)
plot(q[, 1], q[, 3], xlim = c(0, 1), ylim = c(0, 1),
    xlab = expression(q[1]), ylab = expression(q[3]),
    cex.lab = 1.5)
plot(q[, 2], q[, 3], xlim = c(0, 1), ylim = c(0, 1),
    xlab = expression(q[2]), ylab = expression(q[3]),
    cex.lab = 1.5)
```

Histograms


Frequency
0
400
800
0.0
0.4
0.8
q_1
Frequency
0
400
800
0.0
0.4
0.8
q_2
Frequency
0
400
800
0.0
0.4
0.8
q_3

Scatteplots

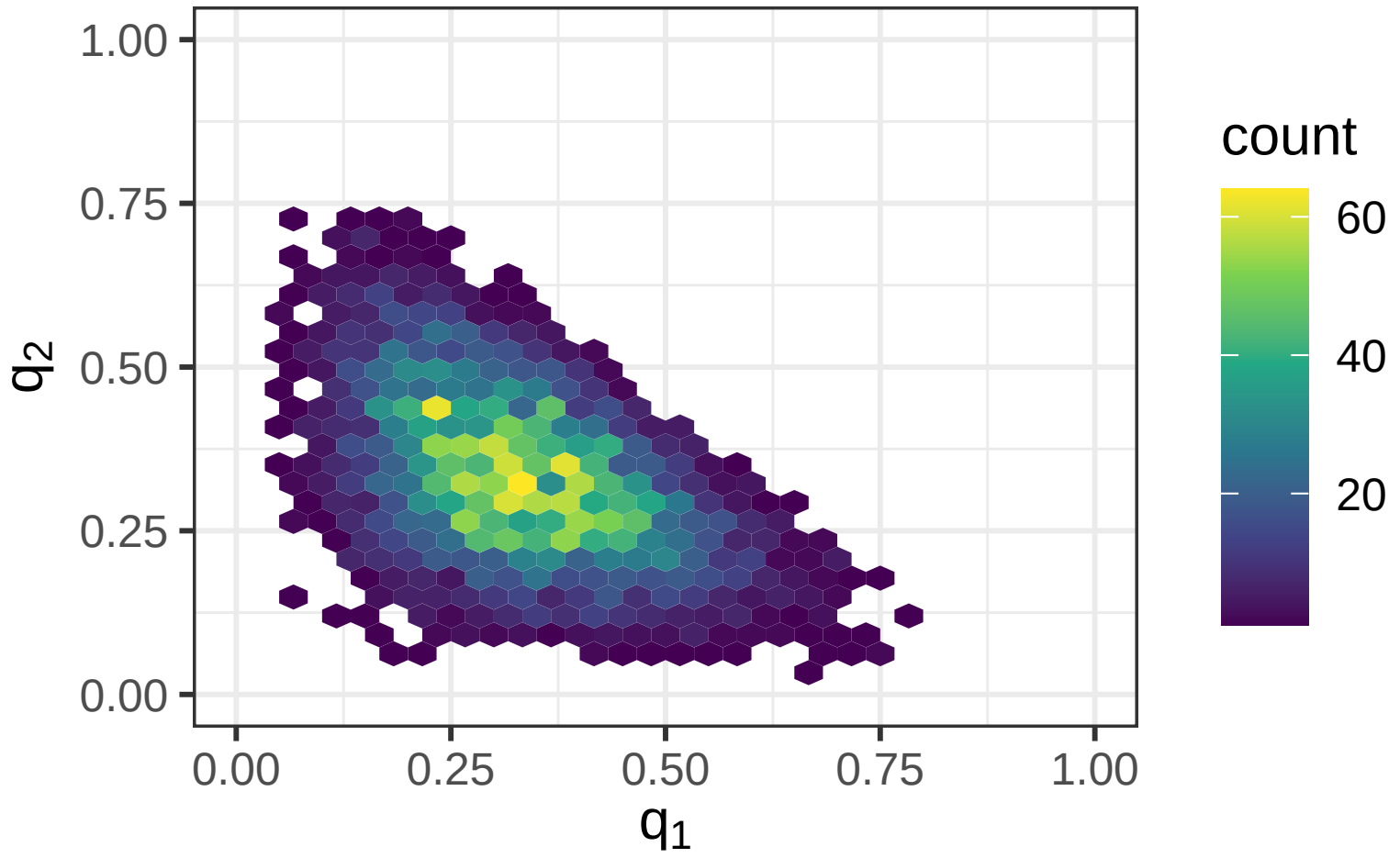


Displaying samples from Dirichlets

```
library(hexbin)
library(ggplot2)
nsim <- 5000
ex1 <- ex2 <- NULL
ex1[1] <- ex1[2] <- ex1[3] <- 5
qex1 <- rdiric(nsim, ex1)
ex2[1] <- 6
ex2[2] <- 4
ex2[3] <- 1
qex2 <- rdiric(nsim, ex2)
# See
# https://www.r-graph-gallery.com/2d-density-plot-with-ggplot2.html
```

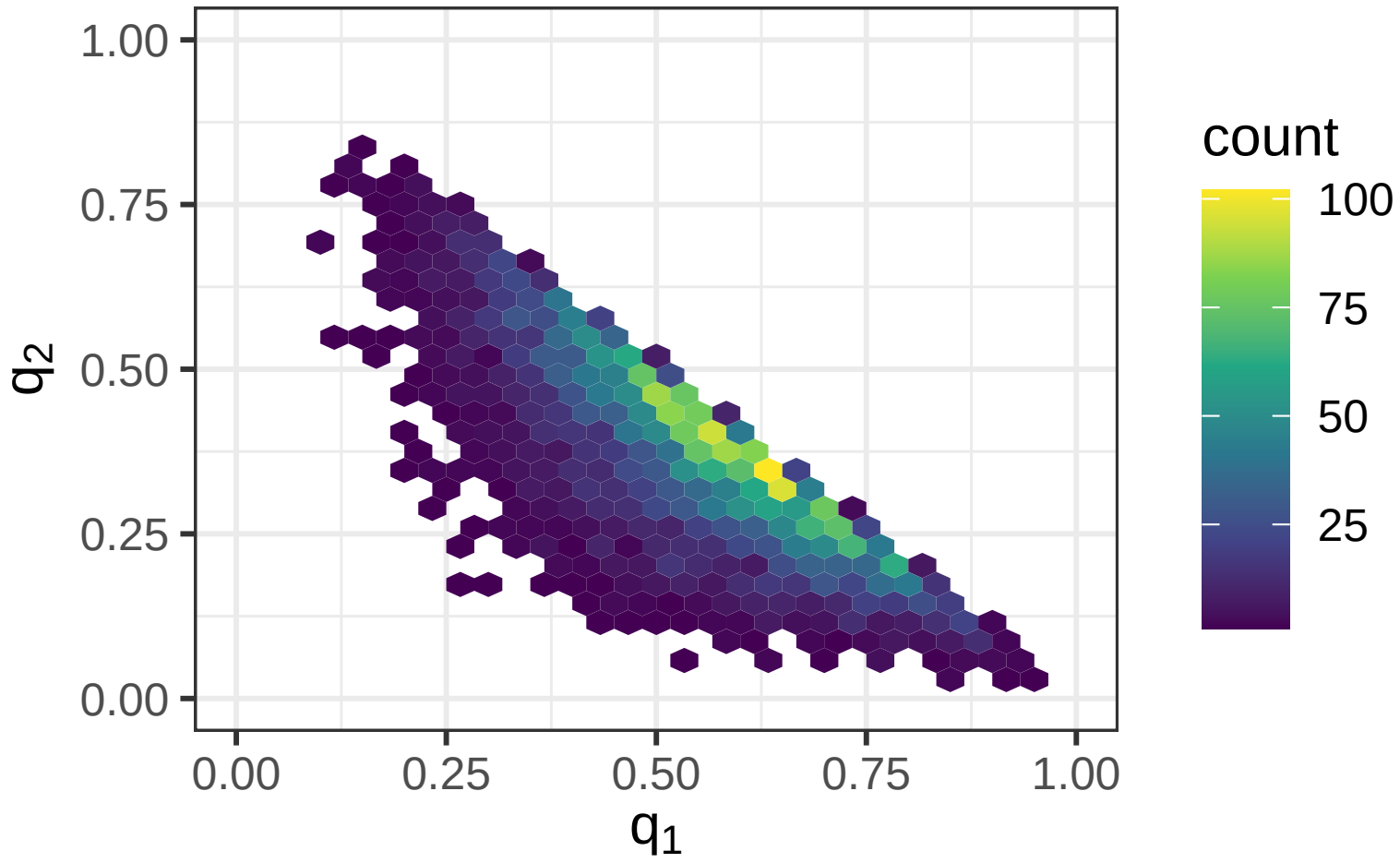
Displaying samples from a Dirichlet(5,5,5)

```
ggplot(data.frame(qex1[, 1], qex1[, 2]), aes(x = qex1[,
    1], y = qex1[, 2])) + geom_hex(bins = 30) + scale_fill_continuous(type = "v
    theme_bw() + xlab(expression(q[1])) + ylab(expression(q[2])) +
    lims(x = c(0, 1), y = c(0, 1))
```



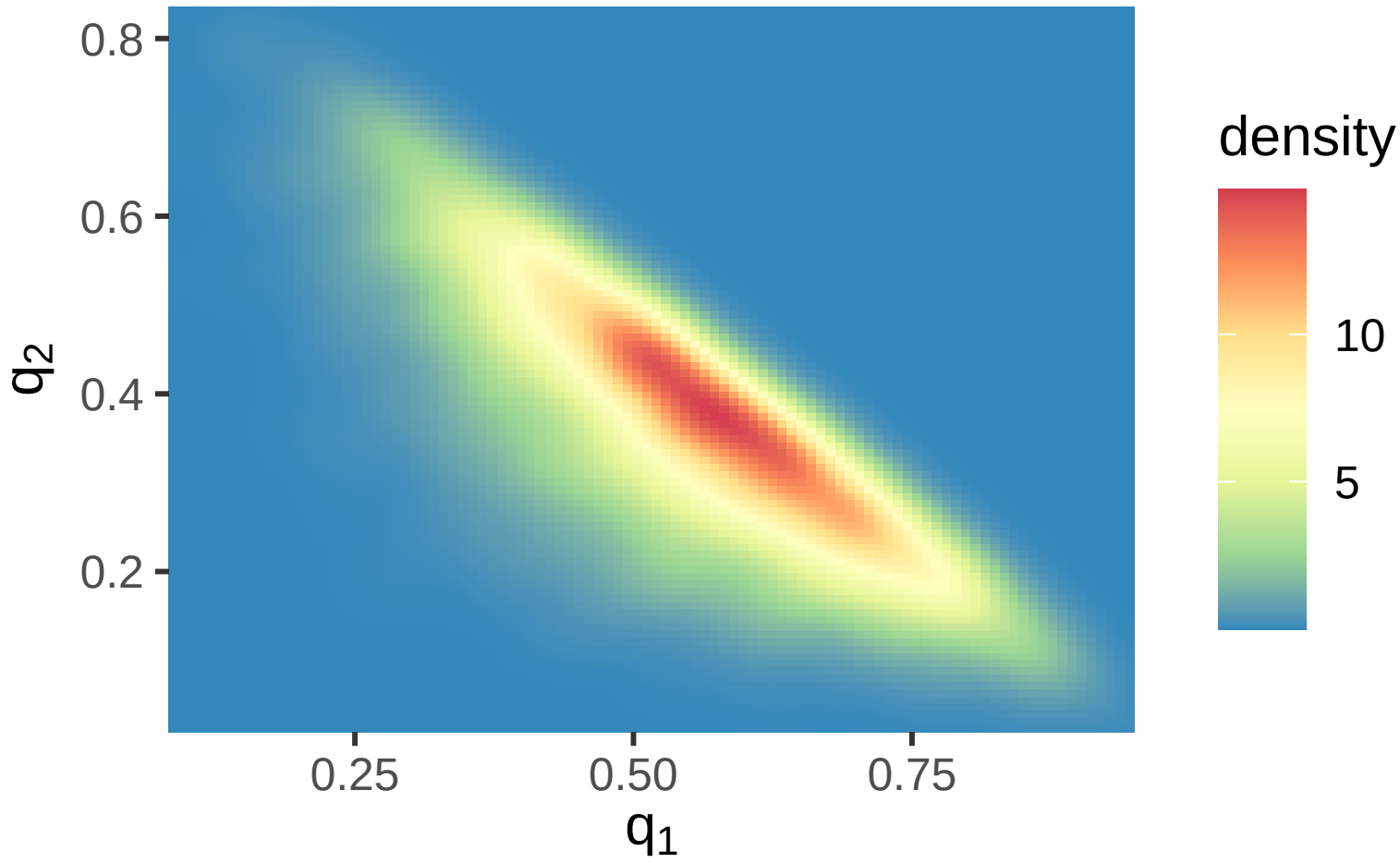
Displaying samples from a Dirichlet(6,4,1)

```
ggplot(data.frame(qex2[, 1], qex2[, 2]), aes(x = qex2[,
    1], y = qex2[, 2])) + geom_hex(bins = 30) + scale_fill_continuous(type = "v
    theme_bw() + xlab(expression(q[1])) + ylab(expression(q[2])) +
    lims(x = c(0, 1), y = c(0, 1))
```



Displaying samples from a Dirichlet(6,4,1)

```
ggplot(data.frame(qex2[, 1], qex2[, 2]), aes(x = qex2[,
    1], y = qex2[, 2])) + stat_density_2d(aes(fill = ..density..),
    geom = "raster", contour = FALSE) + scale_fill_distiller(palette = "Spectral
    direction = -1) + scale_x_continuous(expand = c(0,
    0)) + scale_y_continuous(expand = c(0, 0)) + xlab(expression(q[1])) +
    ylab(expression(q[2]))
```



Functions of interest: implied priors

We assume a "dirichlet(1,1,1)"' distribution

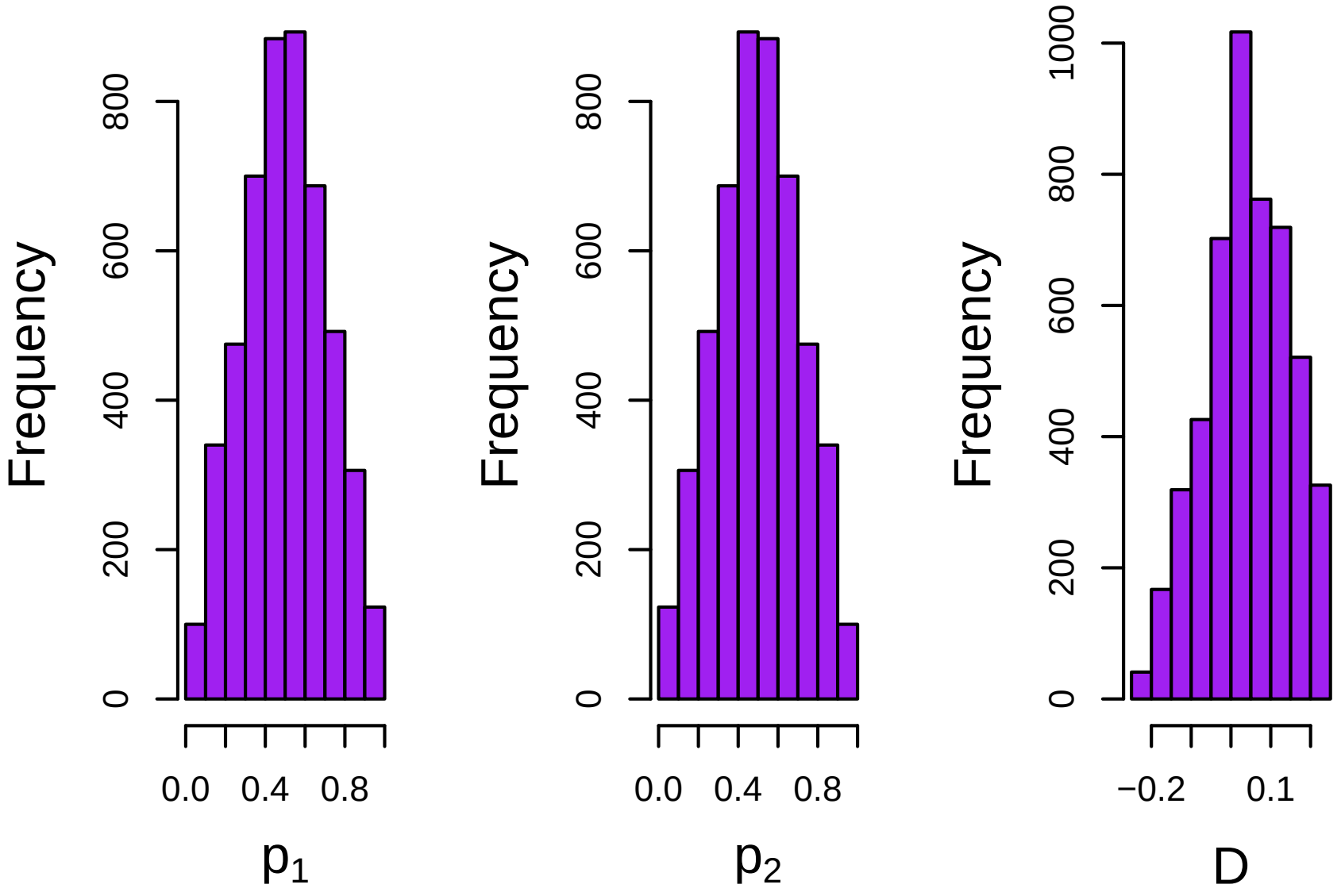
```
q1 <- q[, 1]
q2 <- q[, 2]
q3 <- q[, 3]
p1 <- q1 + q2/2
p2 <- q3 + q2/2
f <- (q1 - p1^2)/(p1 * p2)
D <- q1 - p1^2
psi <- q2^2/(p1 * p2)
## Functions of interest
cat("Prior prob f>0: ", sum(f > 0)/nsim, "\n")
## Prior prob f>0:  0.669
cat("Prior prob D>0: ", sum(D > 0)/nsim, "\n")
## Prior prob D>0:  0.669
```

Functions of interest

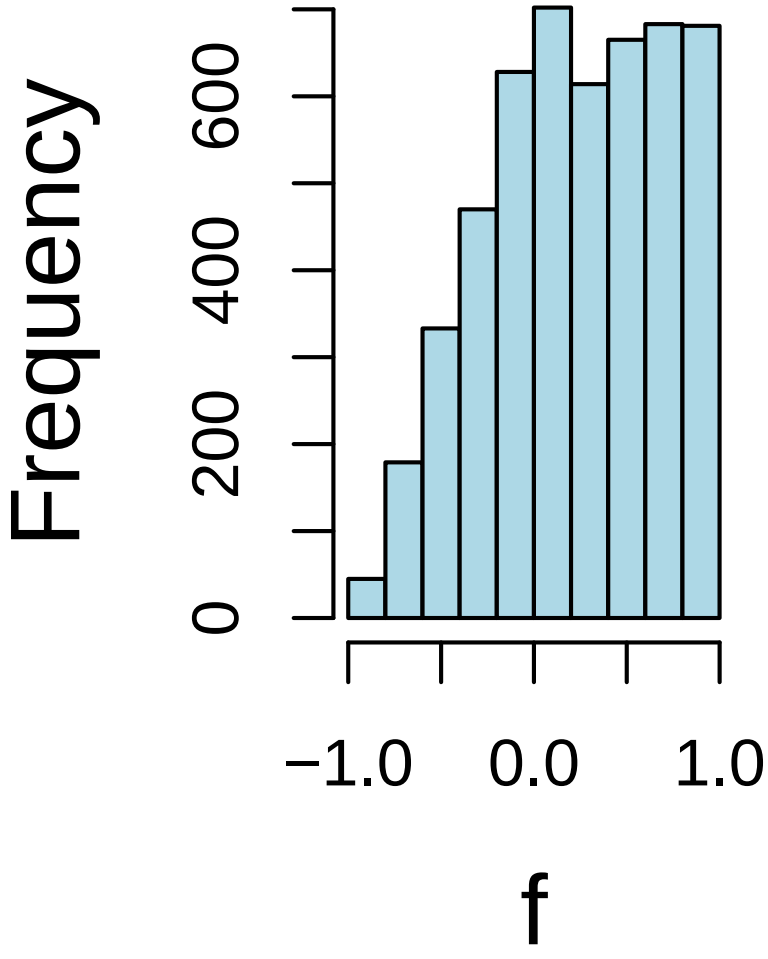
Examine prior summaries for different functions of interest.

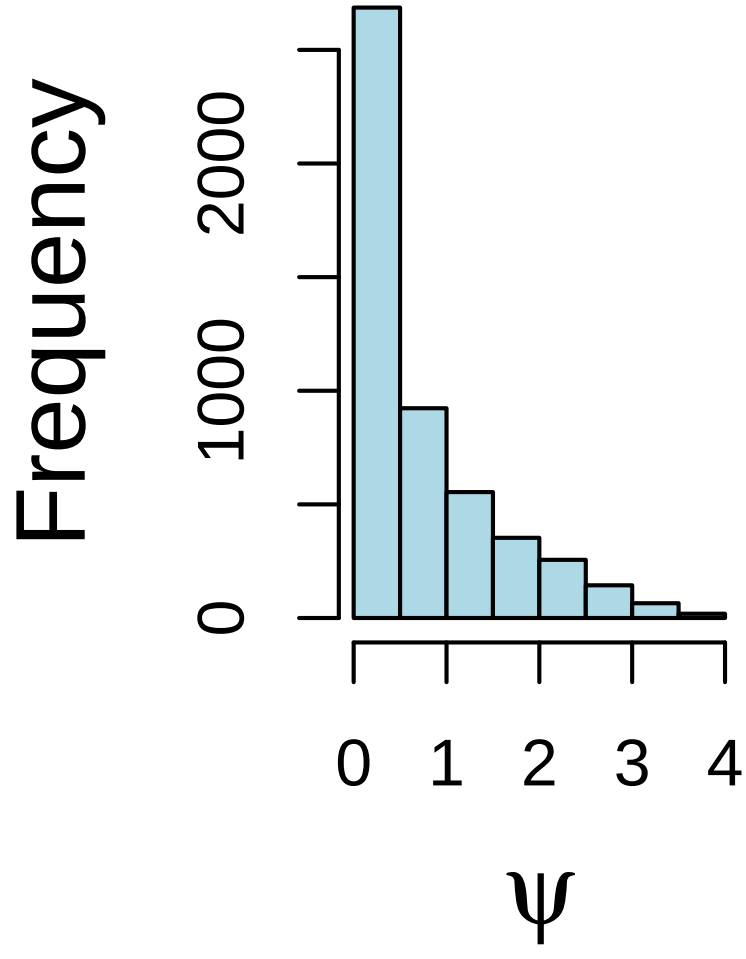
```
par(mfrow = c(1, 3))
hist(p1, main = "", xlab = expression(p[1]), cex.lab = 1.5)
hist(p2, main = "", xlab = expression(p[2]), cex.lab = 1.5)
hist(D, main = "", xlab = expression(D), cex.lab = 1.5)
par(mfrow = c(1, 2))
hist(f, main = "", xlab = "f", cex.lab = 1.5)
hist(psi, main = "", xlab = expression(psi), cex.lab = 1.5)
```

Functions of interest: priors on p_1, p_2, D

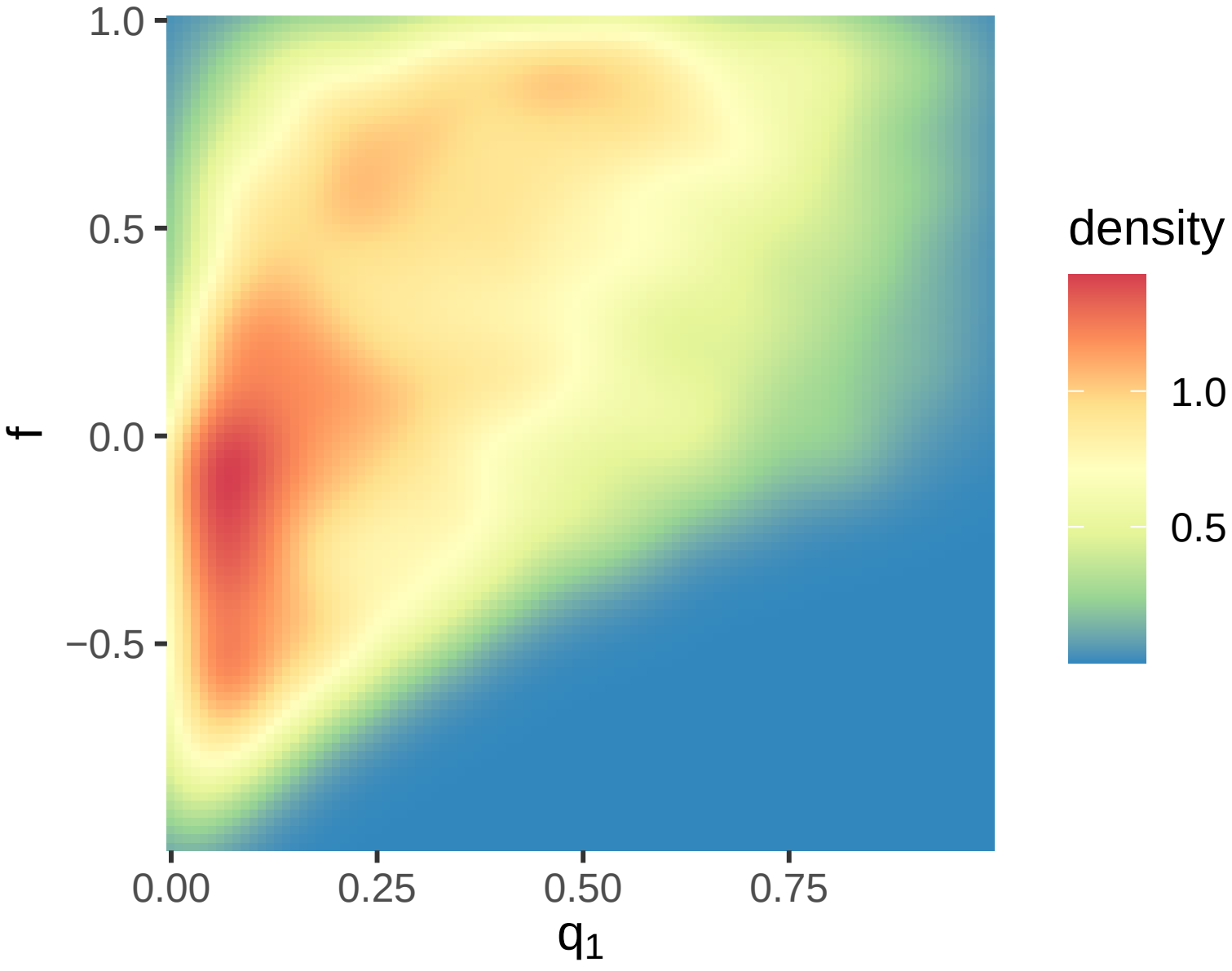


Functions of interest: priors on f, ψ.

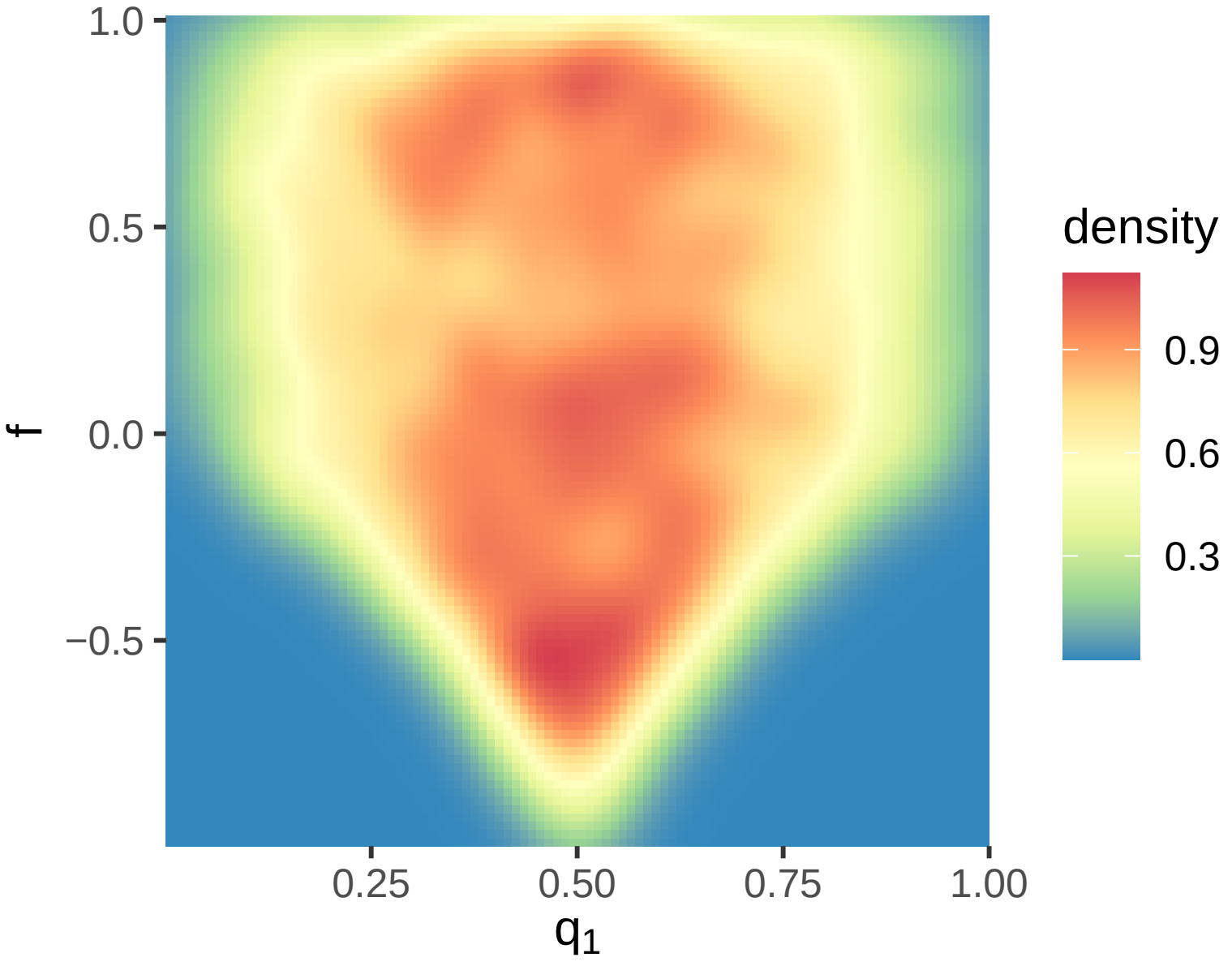




Prior on f and q_1



Prior on f and p_1

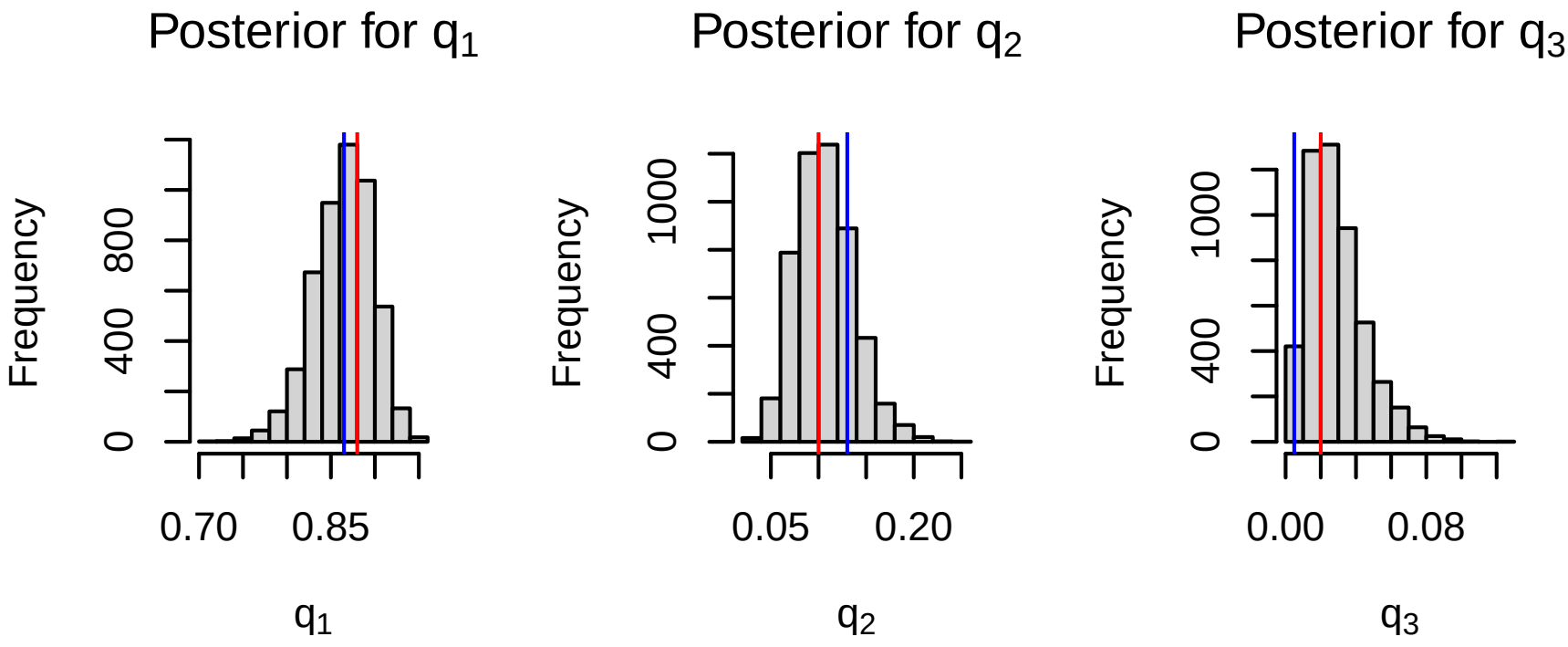


Bayes analysis of (88,10,2) data

```
n1 <- 88
n2 <- 10
n3 <- 2
p1 <- 88/100 + 0.5 * 10/100  # Estimated allele frequencies
p2 <- 2/100 + 0.5 * 10/100  # for A1 and A2
v1 <- v2 <- v3 <- 1
q <- rdiric(nsim, c(n1 + v1, n2 + v2, n3 + v3))  # The posterior
q1 <- q[, 1]
q2 <- q[, 2]
q3 <- q[, 3]
par(mfrow = c(1, 3))
hist(q1, xlab = expression(q[1]), main = expression(paste("Posterior for ",
    q[1])))
abline(v = n1/(n1 + n2 + n3), col = "red")
abline(v = p1^2, col = "blue")
hist(q2, xlab = expression(q[2]), main = expression(paste("Posterior for ",
    q[2])))
abline(v = n2/(n1 + n2 + n3), col = "red")
abline(v = 2 * p1 * p2, col = "blue")
hist(q3, xlab = expression(q[3]), main = expression(paste("Posterior for ",
    q[3])))
abline(v = n3/(n1 + n2 + n3), col = "red")
abline(v = p2^2, col = "blue")
```

Bayes analysis of (88,10,2) data

Univariate posterior distributions: blue lines are the MLEs in the full model, red lines under the HWE model

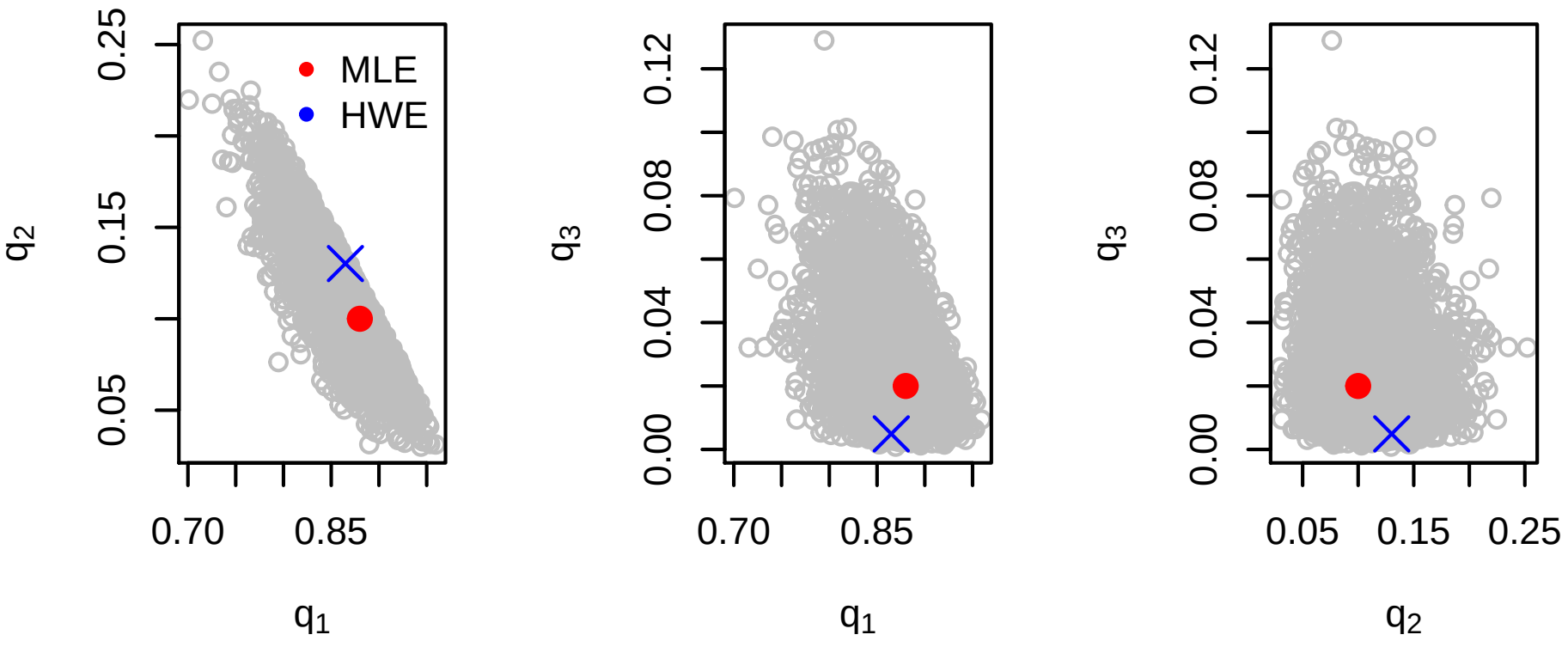


Bayes analysis of (88,10,2) data

```
par(mfrow = c(1, 3))
plot(q2 ~ q1, xlab = expression(q[1]), ylab = expression(q[2]),
    col = "grey")
points(n1/(n1 + n2 + n3), n2/(n1 + n2 + n3), col = "red",
    pch = 20, cex = 2)
points(p1^2, 2 * p1 * p2, col = "blue", pch = 4, cex = 2)
legend("topright", legend = c("MLE", "HWE"), col = c("red",
    "blue"), pch = c(20, 20), bty = "n")
plot(q3 ~ q1, xlab = expression(q[1]), ylab = expression(q[3]),
    col = "grey")
points(n1/(n1 + n2 + n3), n3/(n1 + n2 + n3), col = "red",
    pch = 20, cex = 2)
points(p1^2, p2^2, col = "blue", pch = 4, cex = 2)
plot(q3 ~ q2, xlab = expression(q[2]), ylab = expression(q[3]),
    col = "grey")
points(n2/(n1 + n2 + n3), n3/(n1 + n2 + n3), col = "red",
    pch = 20, cex = 2)
points(2 * p1 * p2, p2^2, col = "blue", pch = 4, cex = 2)
```

Bayes analysis of (88,10,2) data

Bivariate posterior distributions: blue lines are the MLEs in the full model, red lines under the HWE model



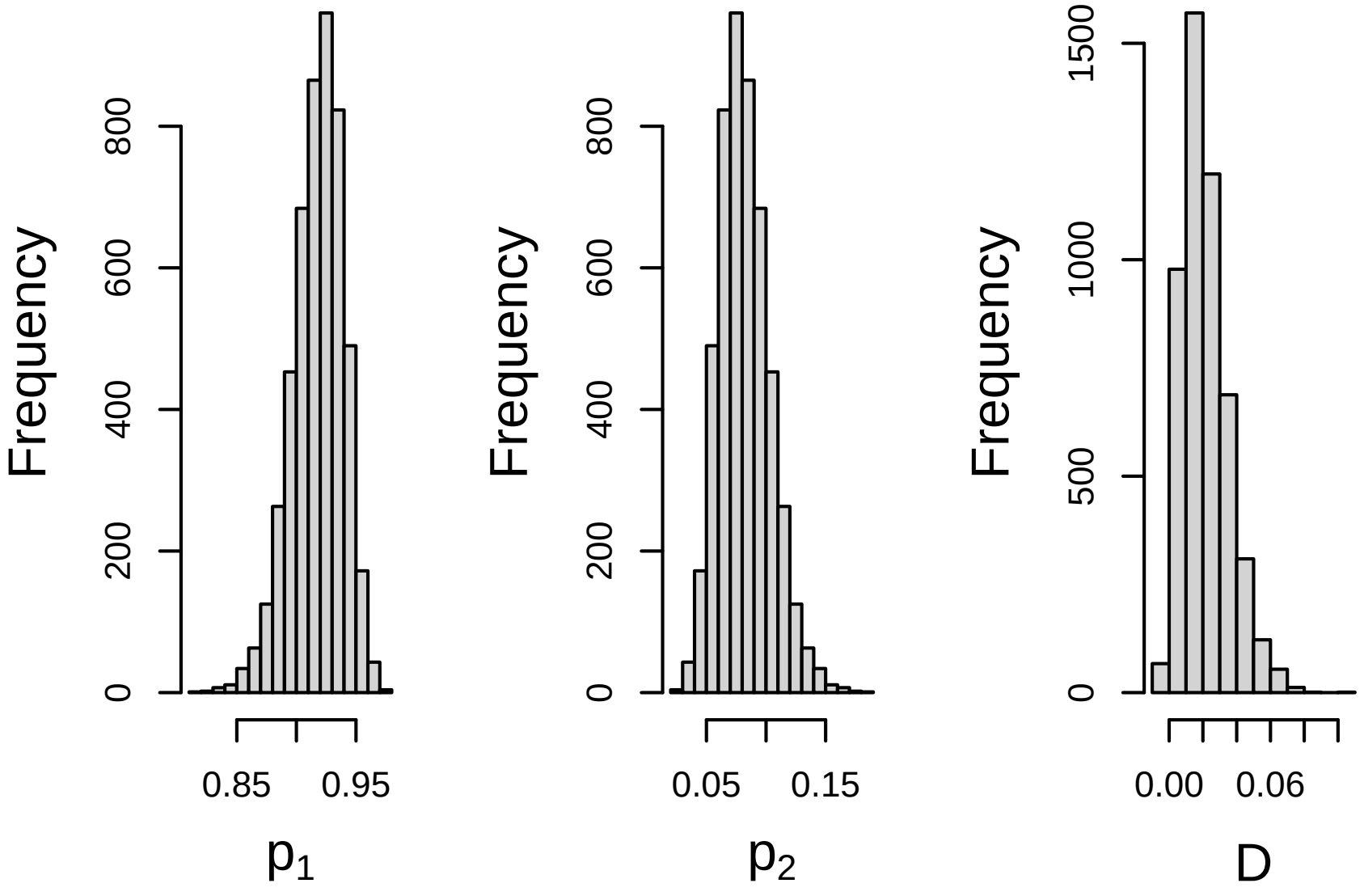
Inference for f

The MLE is $\hat{f} = 0.23$ with asymptotic standard error 0.17.

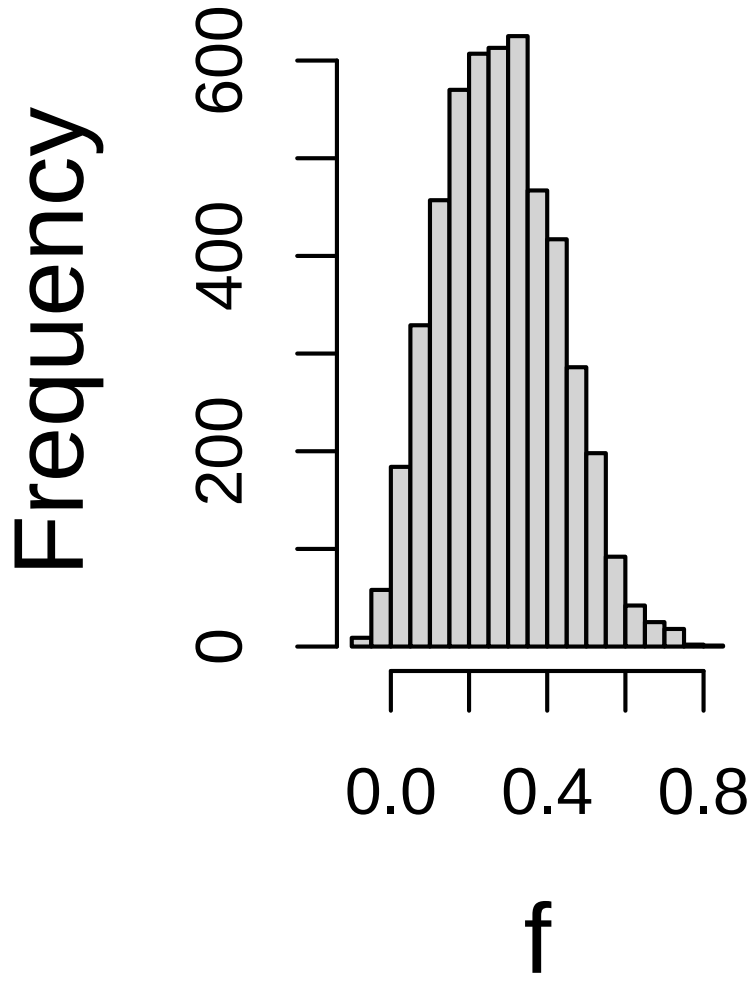
Hence, a 95% asymptotic confidence interval is

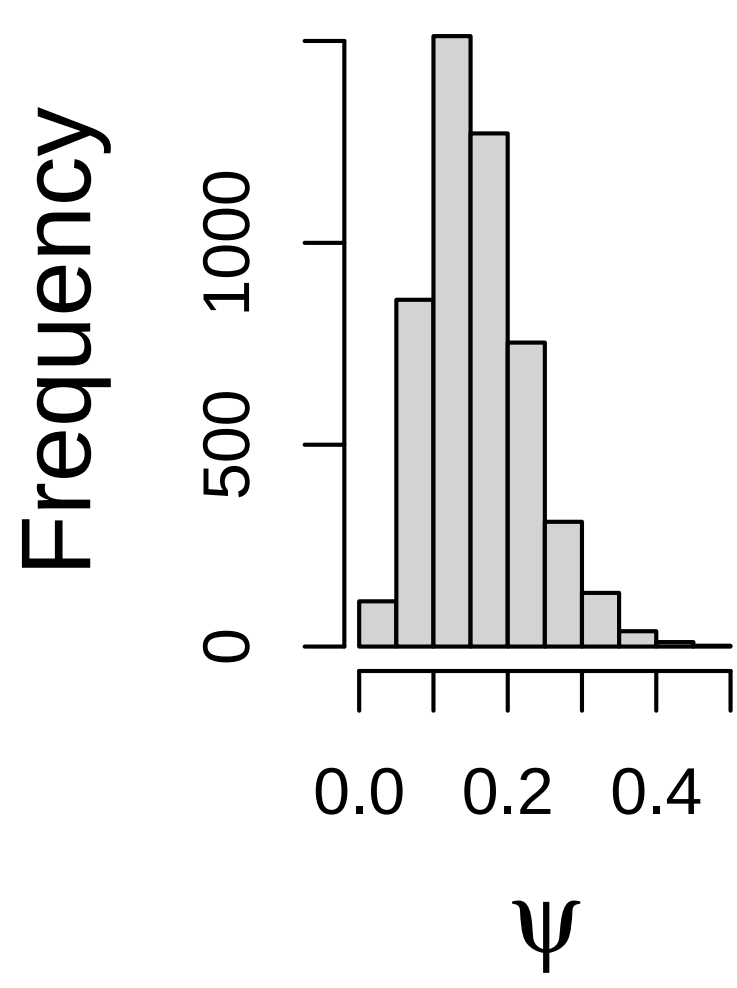
$$(0.23 - 1.96 \times 0.17, 0.23 + 1.96 \times 0.17) = (-0.1032, 0.5632).$$

Posterior summaries



Posterior summaries





HWE analysis via Stan

Stan is not needed here, but useful if we want a non-Dirichlet prior.

In this example we analyze the data under HWE

```
data {
    int y[3];
}
parameters {
    real<lower=0,upper=1> p;
}
transformed parameters {
    simplex[3] theta;
    theta[1] = p*p;
    theta[2] = 2*p*(1-p);
    theta[3] = (1-p)*(1-p);
}
model {
    p~beta(1, 1);
    y~multinomial(theta);
}
```

HWE analysis via Stan

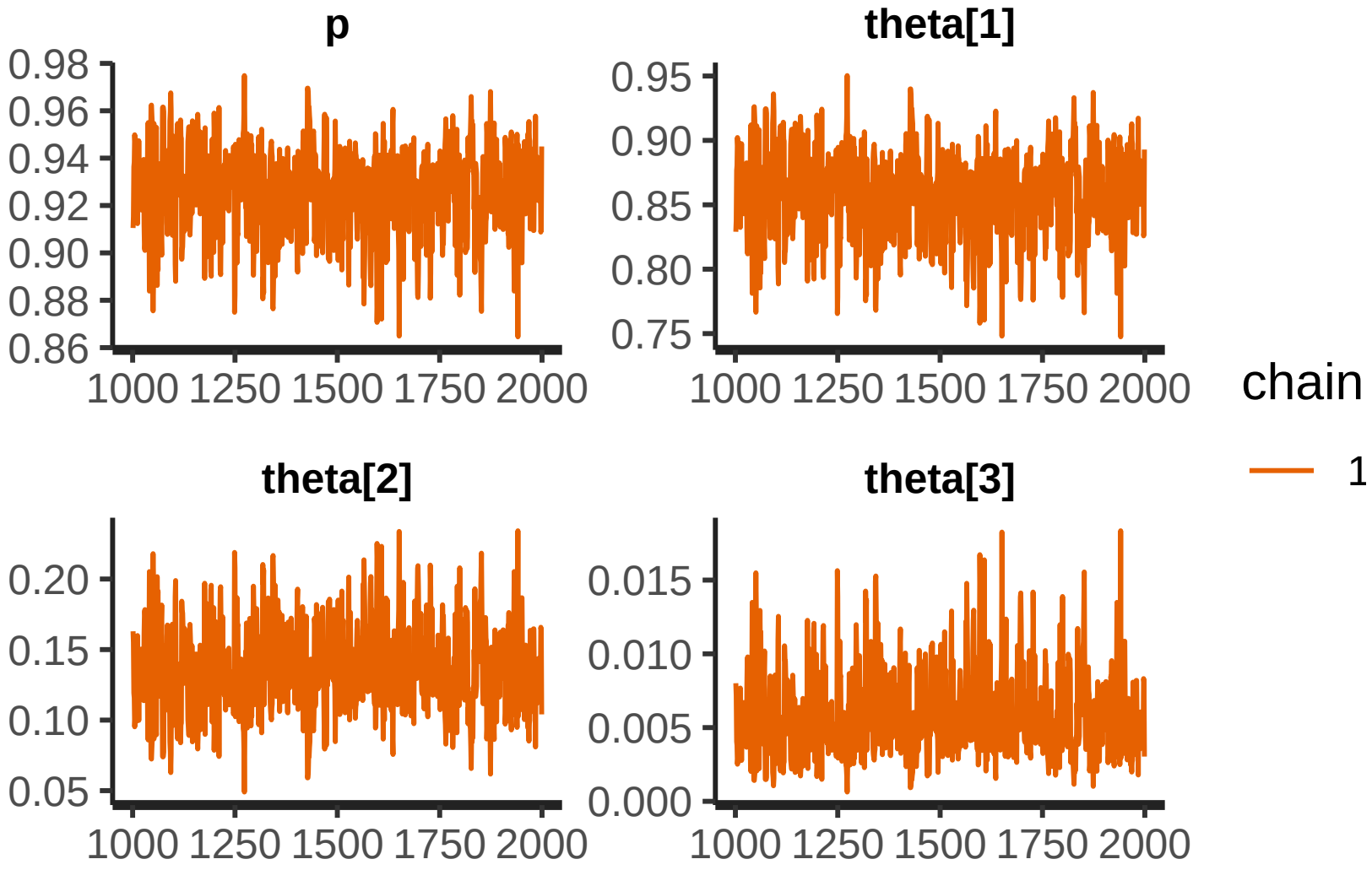
```
library(rstan)
# download.file('http://faculty.washington.edu/kenrice/sisgbayes
# destfile = 'HWEexample.stan')
stanexample <- stan("HWEexample.stan", data = list(y = c(88,
    10, 2)), iter = 2000, chains = 1, seed = 1234)
```

HWE analysis via Stan

```
print(stanexample)
## Inference for Stan model: HWEexample.
## 1 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=1000.
##
##           mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## p         0.93    0.00 0.02   0.89   0.91   0.93   0.94   0.96   375    1
## theta[1]  0.86    0.00 0.03   0.79   0.83   0.86   0.88   0.92   372    1
## theta[2]  0.14    0.00 0.03   0.08   0.12   0.13   0.16   0.20   369    1
## theta[3]  0.01    0.00 0.00   0.00   0.00   0.01   0.01   0.01   409    1
## lp__    -47.00    0.04 0.70 -48.97 -47.18 -46.74 -46.56 -46.50   395    1
##
## Samples were drawn using NUTS(diag_e) at Tue Jul 21 09:57:31 2020.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

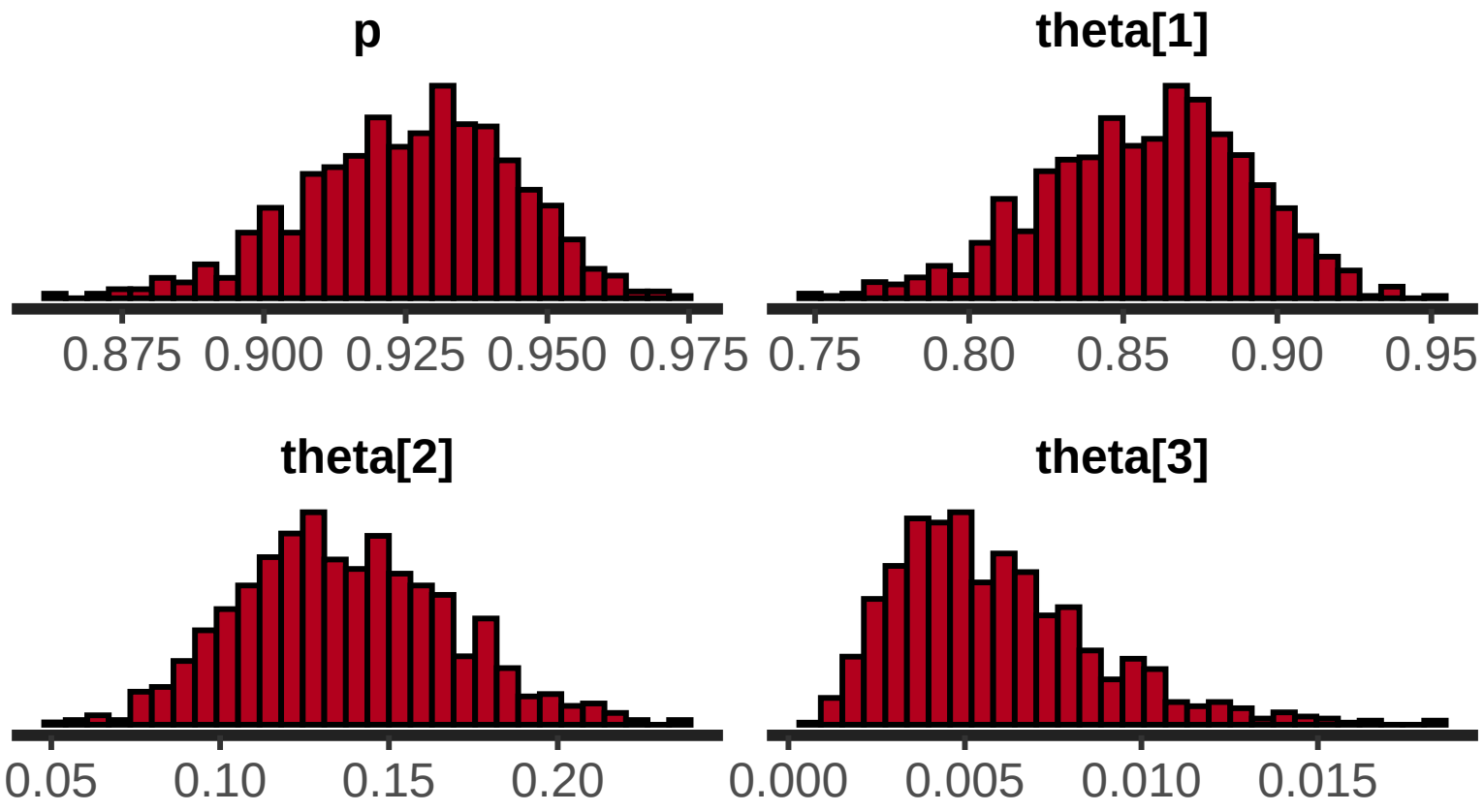
HWE analysis via Stan

```
traceplot(stanexample)
```



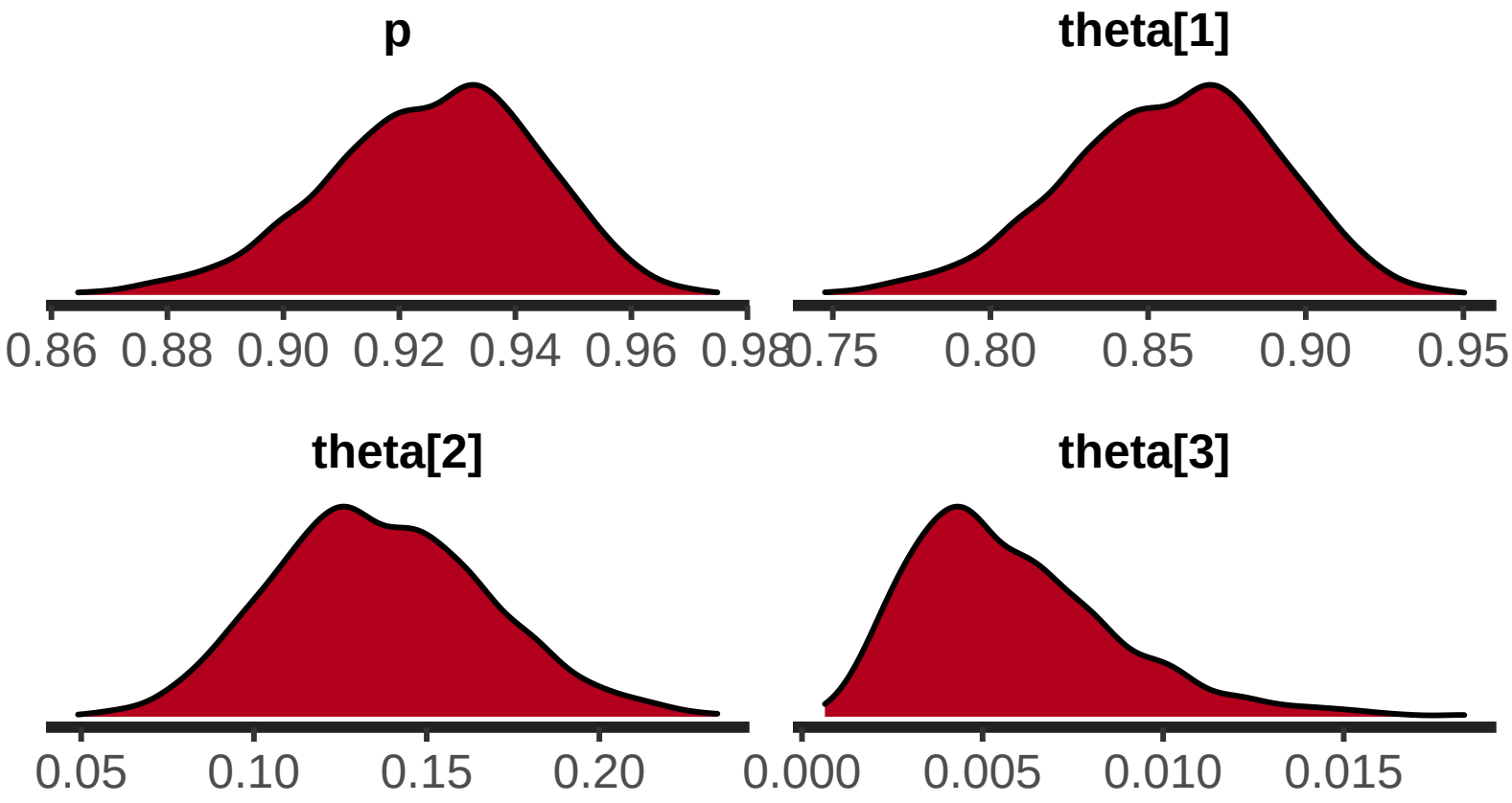
HWE analysis via Stan

```
stan_hist(stanexample)
```



HWE analysis via Stan

```
stan_dens(stanexample)
```



HWE analysis via Stan

Now run a second example with the saturated (i.e, non-HWE) model

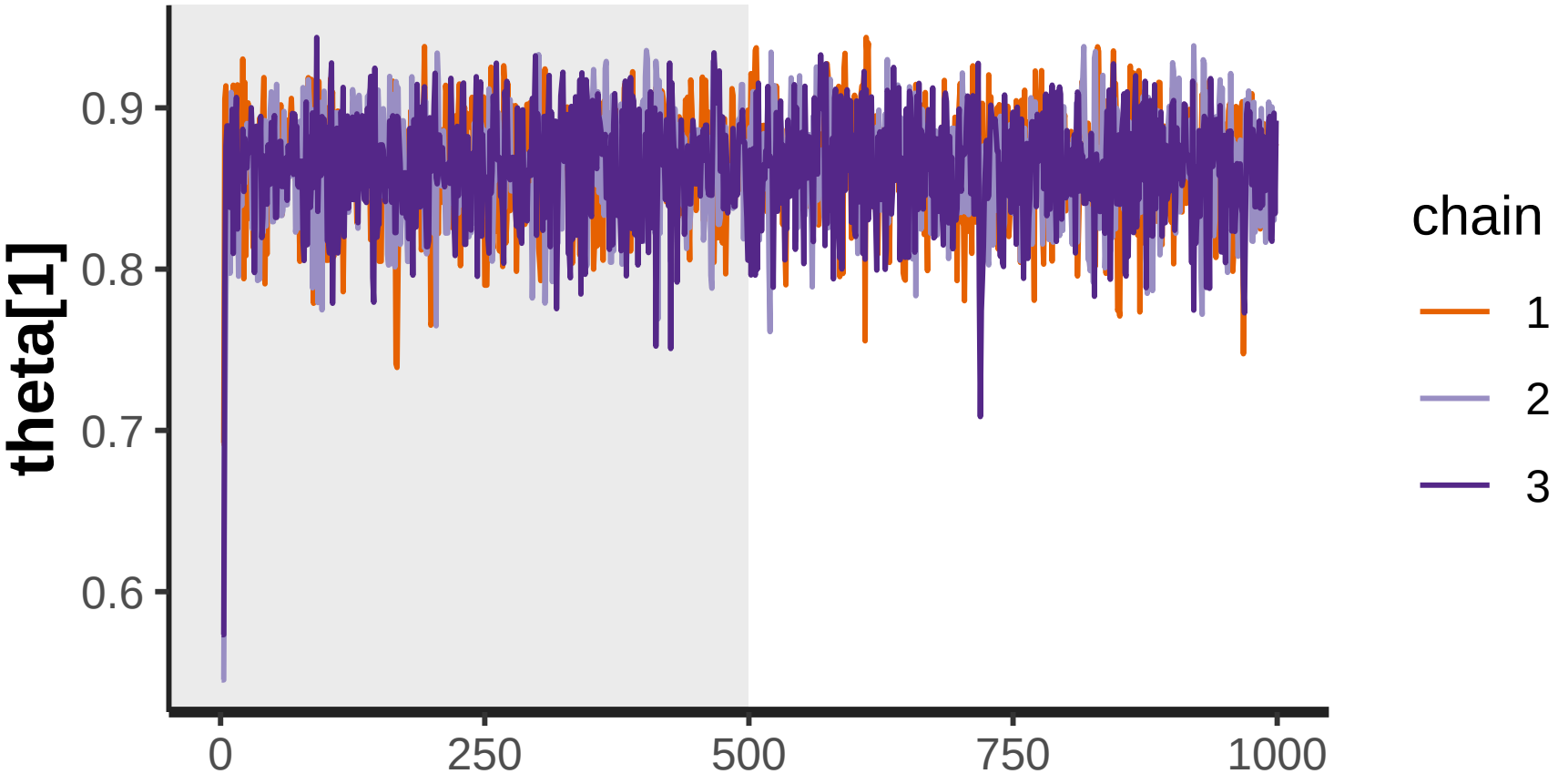
```
# download.file('http://faculty.washington.edu/kenrice/sisgbayes
# destfile = 'HWEexampleSaturated.stan')
stanexample2 <- stan("HWEexampleSaturated.stan",
    data = list(y = c(88, 10, 2), alpha = c(1,
        1, 1)), iter = 1000, chains = 3,
    seed = 1234)
```

HWE analysis via Stan

```
summary(stanexample2)
## $summary
##                  mean      se_mean          sd         2.5%          25%
## theta[1]   0.86360833 0.0009177701 0.03247813   0.79596579   0.84183473
## theta[2]   0.10652837 0.0008128355 0.02915368   0.05380556   0.08605319
## theta[3]   0.02986331 0.0004905329 0.01599303   0.00706339   0.01849235
## lp__     -49.16560214 0.0368739501 0.95154244 -51.73965844 -49.50731106
##                  50%          75%        97.5%     n_eff      Rhat
## theta[1]   0.86603690   0.88595990   0.92127942 1252.3166 0.9996181
## theta[2]   0.10415845   0.12591945   0.16993932 1286.4157 1.0009976
## theta[3]   0.02697804   0.03861392   0.06708907 1062.9802 0.9987465
## lp__     -48.87524868 -48.49017587 -48.24035755  665.9122 1.0029787
##
## $c_summary
## , , chains = chain:1
##
##           stats
## parameter          mean         sd          2.5%          25%         50%
##   theta[1]   0.86544156 0.03336252   0.795503620   0.84356470   0.8675657
##   theta[2]   0.10469497 0.03007782   0.052087514   0.08332073   0.1023841
##   theta[3]   0.02986347 0.01592749   0.006617708   0.01807458   0.0270443
##   lp__     -49.21061144 1.02242388 -52.259680694 -49.55879692 -48.8743481
##           stats
## parameter           75%        97.5%
##   theta[1]   0.88878633   0.92535809
##   theta[2]   0.12477818   0.16927576
##   theta[3]   0.03837428   0.06569151
##   lp__     -48.52217595 -48.24097040
##
## , , chains = chain:2
##
##           stats
## parameter          mean         sd          2.5%          25%         50%
##   theta[1]   0.86272825 0.03055311   0.802309589   0.84219761   0.86412279
##   theta[2]   0.10760106 0.02701235   0.056463668   0.08906929   0.10617940
##   theta[3]   0.02967068 0.01561118   0.006680282   0.01863920   0.02725406
```

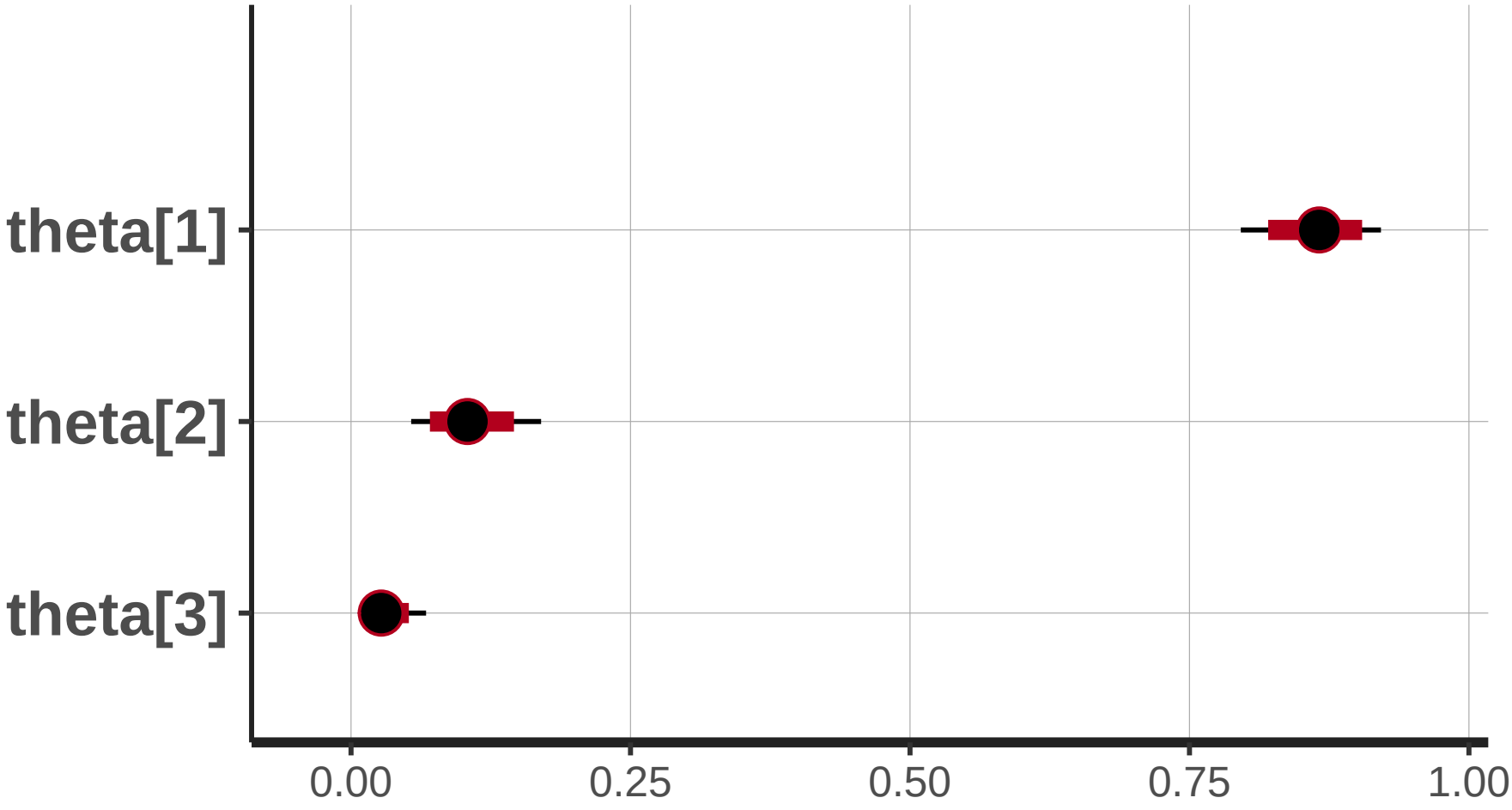
HWE analysis via Stan

```
traceplot(stanexample2, pars = c("theta[1]"), inc_warmup = TRUE)
```



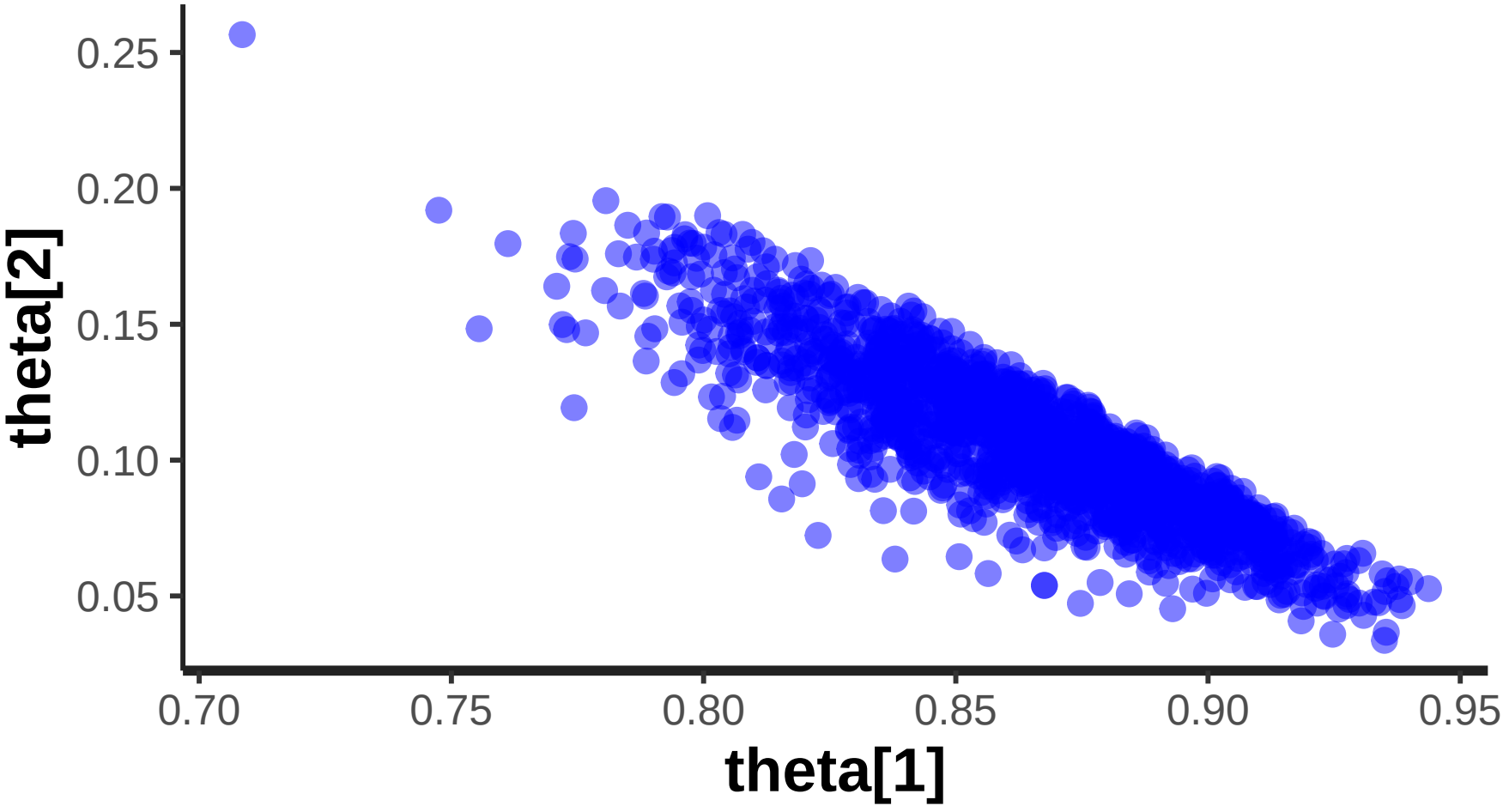
HWE analysis via Stan

```
plot(stanexample2, color = "green")
```



HWE analysis via Stan

```
stan_scat(stanexample2, pars = c("theta[1]", "theta[2]"),
    color = "blue", size = 2)
```



Bayes factor

```
# install.packages('HWEBayes', repos =
# 'http://cran.us.r-project.org')
library(HWEBayes)
bvec0 <- c(1, 1)
bvec1 <- c(1, 1, 1)
nvec <- c(88, 10, 2)
PrnH0 <- DirichNormHWE(nvec, bvec0)
PrnH1sat <- DirichNormSat(nvec, bvec1)
BFH0H1sat <- PrnH0/PrnH1sat
cat("H0 norm = ", PrnH0, "\n")
## H0 norm =  0.0002993684
cat("H1 (sat) norm = ", PrnH1sat, "\n")
## H1 (sat) norm =  0.0001941371
cat("Bayes factor in favor of the null = ", BFH0H1sat,
    "\n")
## Bayes factor in favor of the null =  1.542047
```

Exercises

- Repeat the analyses in these notes for the Lidicker et al data discussed in class, ie with n1=37, n2=20, n3=7. In particular,
  - With a Dirichlet(1,1,1) prior on the 3 probabilities, generate samples, and examine the various summaries, including marginal allele frequencies and the inbreeding coefficient f
  - Evaluate the Bayes factor comparing HWE with the saturated alternative
  - Run the `Stan` examples