2021 SISG Bayesian Statistics for Genetics R Notes: Multinomial and Poisson Count Data

Jon Wakefield
Departments of Statistics and Biostatistics, University of Washington

2021-07-15

Poisson Data

AIDS Data

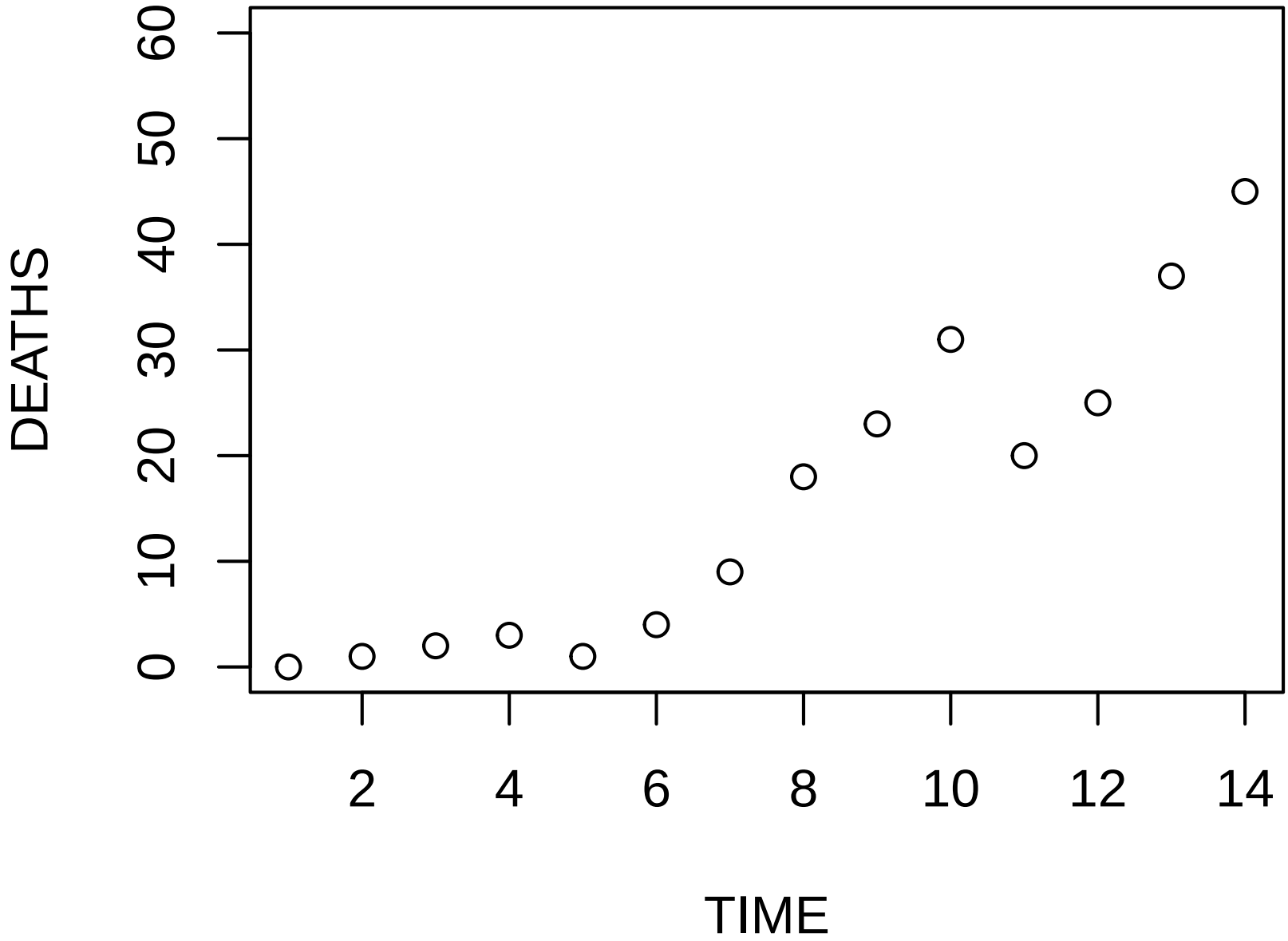
Whyte et al. (1987) reported deaths due to AIDS in Australian 3-month periods from January 1983 to June 1986.

We illustrate Bayesian modeling of these count data using a very simple loglinear model:

$$\begin{aligned} Y_i|\mu_i &\sim \text{Poisson}(\mu_i) \\ \log \mu_i &= \beta_0 + \beta_1 \log(\text{time}_i) \end{aligned}$$

```
library(brinla)
library(dplyr)
library(tidyr)
# Plot the data
data(AIDS, package = "brinla")
head(AIDS, n = 3)
##   TIME DEATHS
## 1    1      0
## 2    2      1
## 3    3      2
```

AIDS Data



AIDS Data

```
# frequentist method
AIDS.glm1 <- glm(DEATHS ~ log(TIME), family = poisson(),
    data = AIDS)
# summary(AIDS.glm1)
round(coef(summary(AIDS.glm1)), 4)
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.9442     0.5116 -3.8003    1e-04
## log(TIME)     2.1748     0.2150 10.1130    0e+00
```

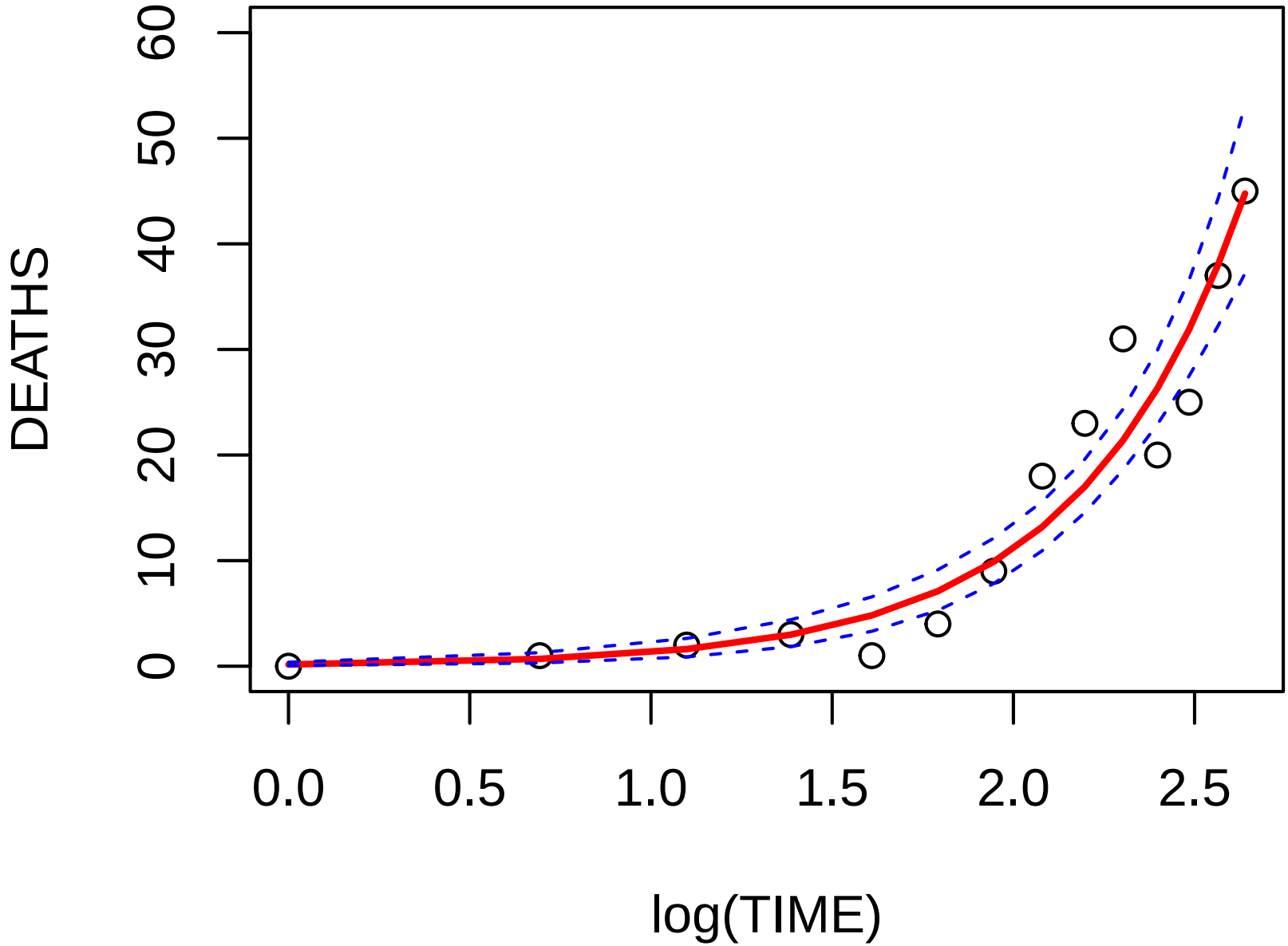
AIDS Data

```
# INLA method
AIDS.inla1 <- inla(DEATHS ~ log(TIME), data = AIDS,
    family = "poisson", control.predictor = list(compute = TRUE))
# Output Posterior Estimates
round(AIDS.inla1$summary.fixed[, 1:5], 4)
##                mean     sd 0.025quant 0.5quant 0.975quant
## (Intercept) -1.9429 0.5112    -2.9902  -1.9275    -0.9829
## log(TIME)    2.1749 0.2149     1.7687   2.1693     2.6132
```

AIDS Data

```
# Plot the results
plot(DEATHS ~ log(TIME), data = AIDS, ylim = c(0, 60))
lines(log(AIDS$TIME), AIDS.inla1$summary.fitted.values$mean,
    lwd = 2, col = "red")
lines(log(AIDS$TIME), AIDS.inla1$summary.fitted.values$"0.025quant",
    lwd = 1, lty = 2, col = "blue")
lines(log(AIDS$TIME), AIDS.inla1$summary.fitted.values$"0.975quant",
    lwd = 1, lty = 2, col = "blue")
```

AIDS Data



AIDS Data: Overdispersion?

The Poisson model is restrictive so we allow for excess-Poisson variation.

```
# frequentist method
AIDS.glmq <- glm(DEATHS ~ log(TIME), family = quasipoisson(),
    data = AIDS)
# summary(AIDS.glmq)
round(coef(summary(AIDS.glmq)), 4)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.9442     0.5905 -3.2923   0.0064
## log(TIME)     2.1748     0.2482  8.7612   0.0000
# Compare with Poisson:
round(coef(summary(AIDS.glm1)), 4)
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.9442     0.5116 -3.8003    1e-04
## log(TIME)     2.1748     0.2150 10.1130    0e+00
```

Overdispersion is estimated as 1.33 and so standard errors are $\sqrt{1.33} = 1.15$ larger in quasipoisson analysis.

AIDS Data

We fit a negative binomial model using frequentist likelihood inference.

```
library(MASS)
AIDS.glm2 <- glm.nb(DEATHS ~ log(TIME), data = AIDS)
round(coef(summary(AIDS.glm2)), 4)
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.0089     0.5527 -3.6349    3e-04
## log(TIME)     2.2038     0.2357  9.3500    0e+00
```

AIDS Data

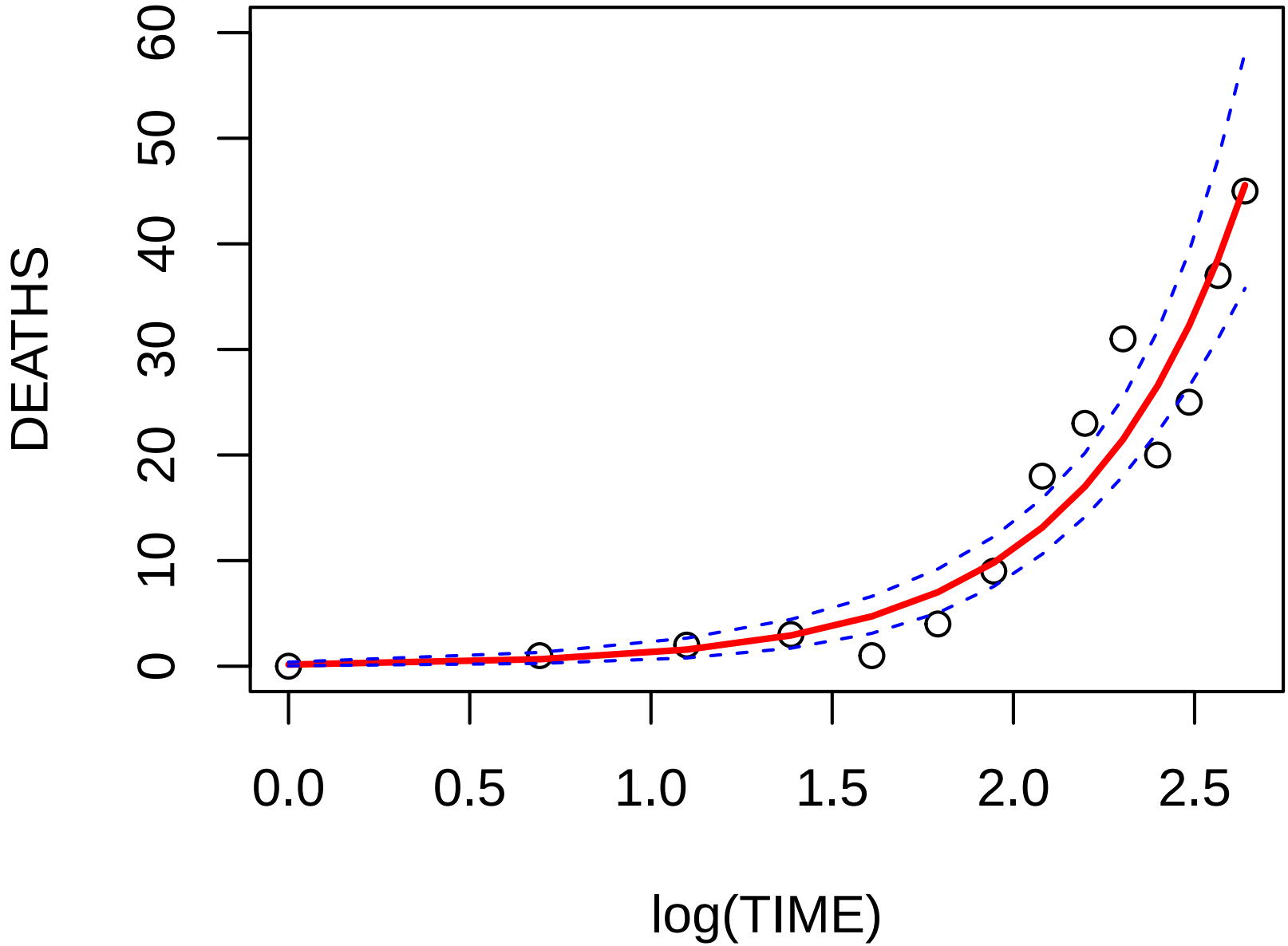
Not fir the negative binomial model using INLA.

```
AIDS.inla2 <- inla(DEATHS ~ log(TIME), data = AIDS,
    family = "nbinomial", control.predictor = list(compute = TRUE))
# Output Posterior Estimates
round(AIDS.inla2$summary.fixed[, 1:5], 4)
##                mean     sd 0.025quant 0.5quant 0.975quant
## (Intercept) -2.0210 0.5784    -3.2318  -1.9965    -0.9527
## log(TIME)    2.2101 0.2491     1.7485   2.1998     2.7310
round(AIDS.inla2$summary.hyperpar[, 1:5], 4)
##                                                                  mean       sd
## size for the nbinomial observations (1/overdispersion) 34891.1 646554.8
##                                                                 0.025quant 0.5quant
## size for the nbinomial observations (1/overdispersion)     8.0802  85.6056
##                                                                 0.975quant
## size for the nbinomial observations (1/overdispersion)    49364.7
```

AIDS Data

```
# Plot the results
plot(DEATHS ~ log(TIME), data = AIDS, ylim = c(0, 60))
lines(log(AIDS$TIME), AIDS.inla2$summary.fitted.values$mean,
    lwd = 2, col = "red")
lines(log(AIDS$TIME), AIDS.inla2$summary.fitted.values$"0.025quant",
    lwd = 1, lty = 2, col = "blue")
lines(log(AIDS$TIME), AIDS.inla2$summary.fitted.values$"0.975quant",
    lwd = 1, lty = 2, col = "blue")
```

AIDS Data



Multinomial Data

Hardy-Weinberg via Fisher's exact test

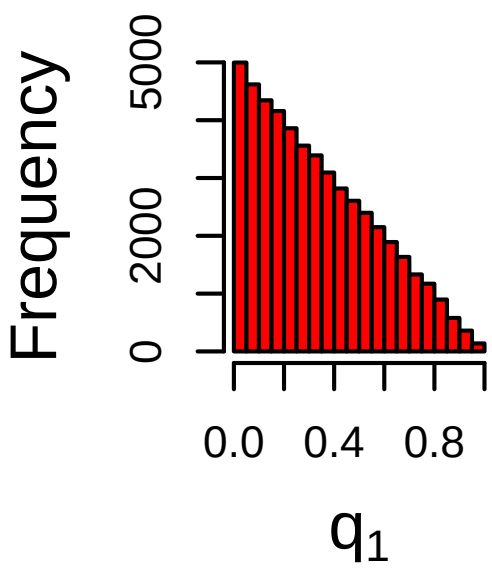
```
library(hwde)
n1 <- 88
n2 <- 10
n3 <- 2
exact <- hwexact(n1, n2, n3)
exact
## [1] 0.06544427
```

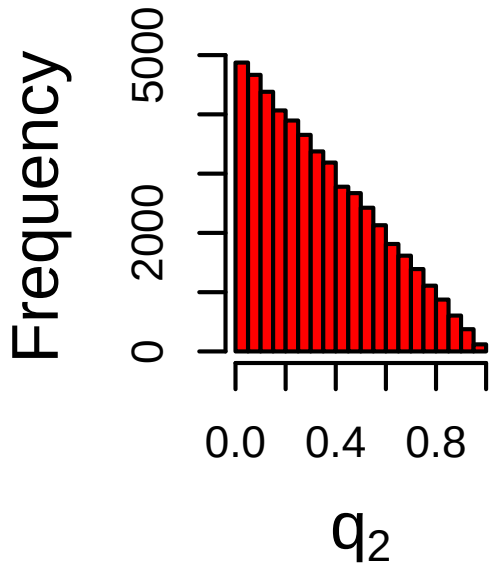
We obtain a p-value of 0.07

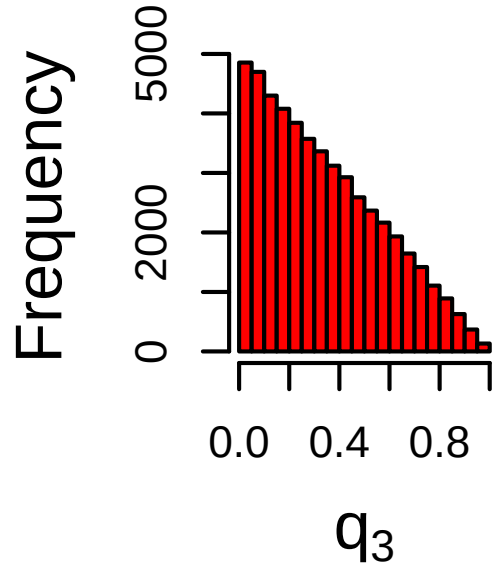
Displaying samples from a Dirichlet(1,1,1)

```
library(VGAM)  # To access the rdiric function
nsim <- 50000
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)
```

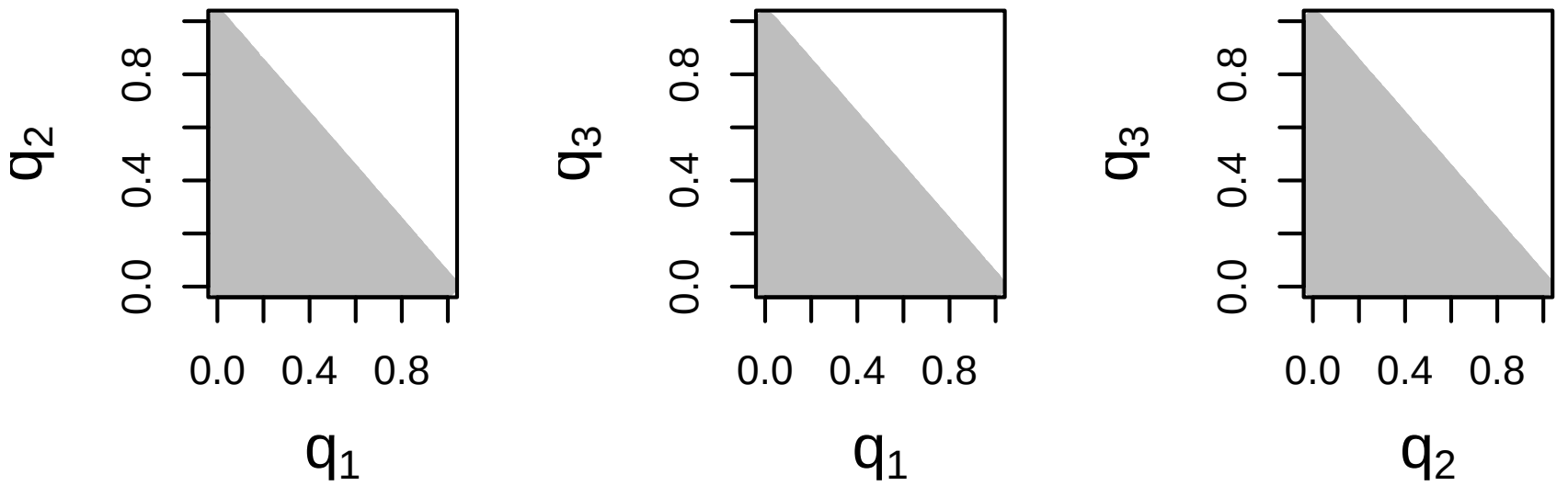
Histograms


Frequency
0
2000
5000
0.0
0.4
0.8
q_1


Frequency
0
2000
5000
0.0
0.4
0.8
q_2


Frequency
0
2000
5000
0.0
0.4
0.8
q_3

Scatteplots

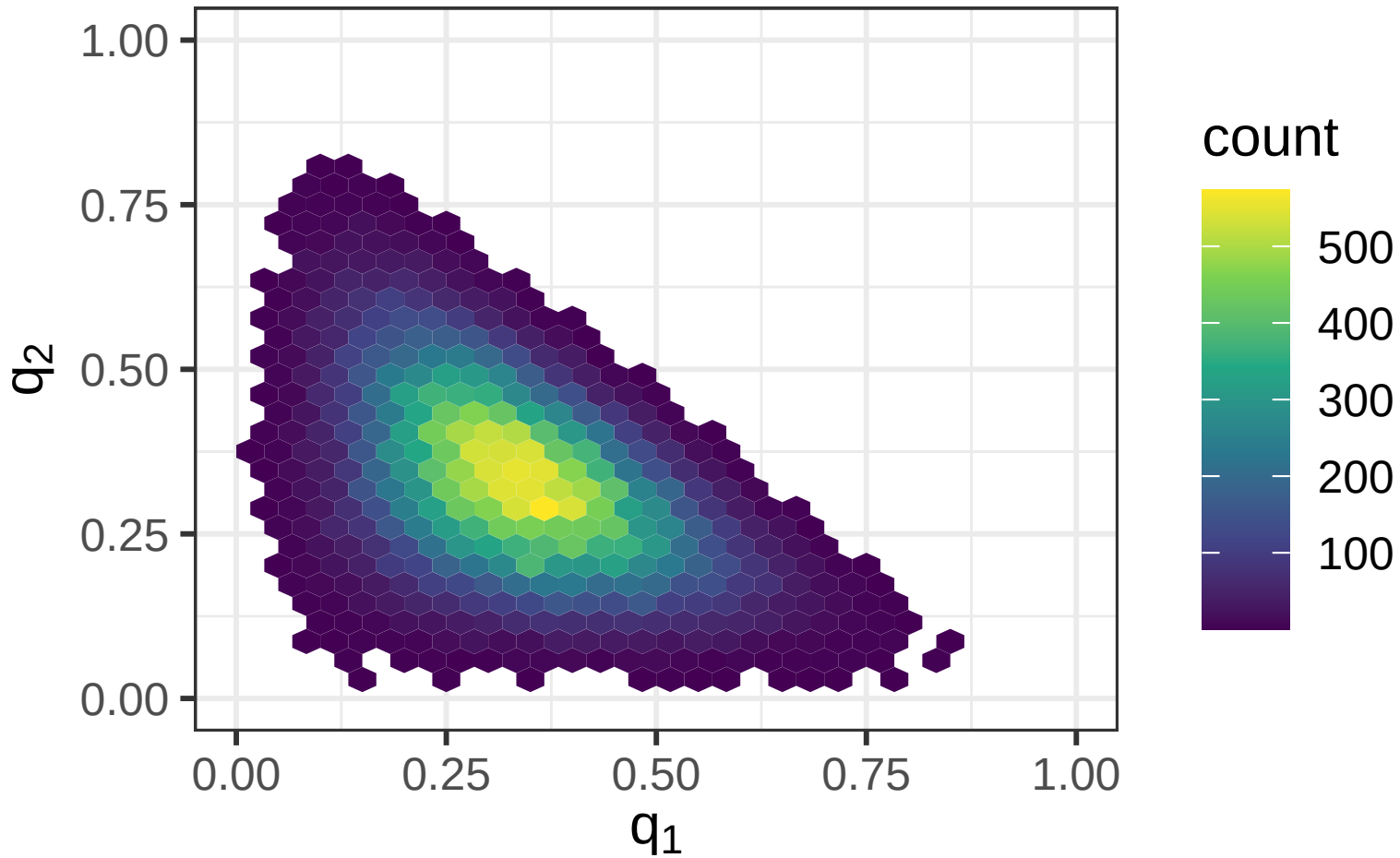


Displaying samples from Dirichlets

```
library(hexbin)
library(ggplot2)
nsim <- 50000
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
```

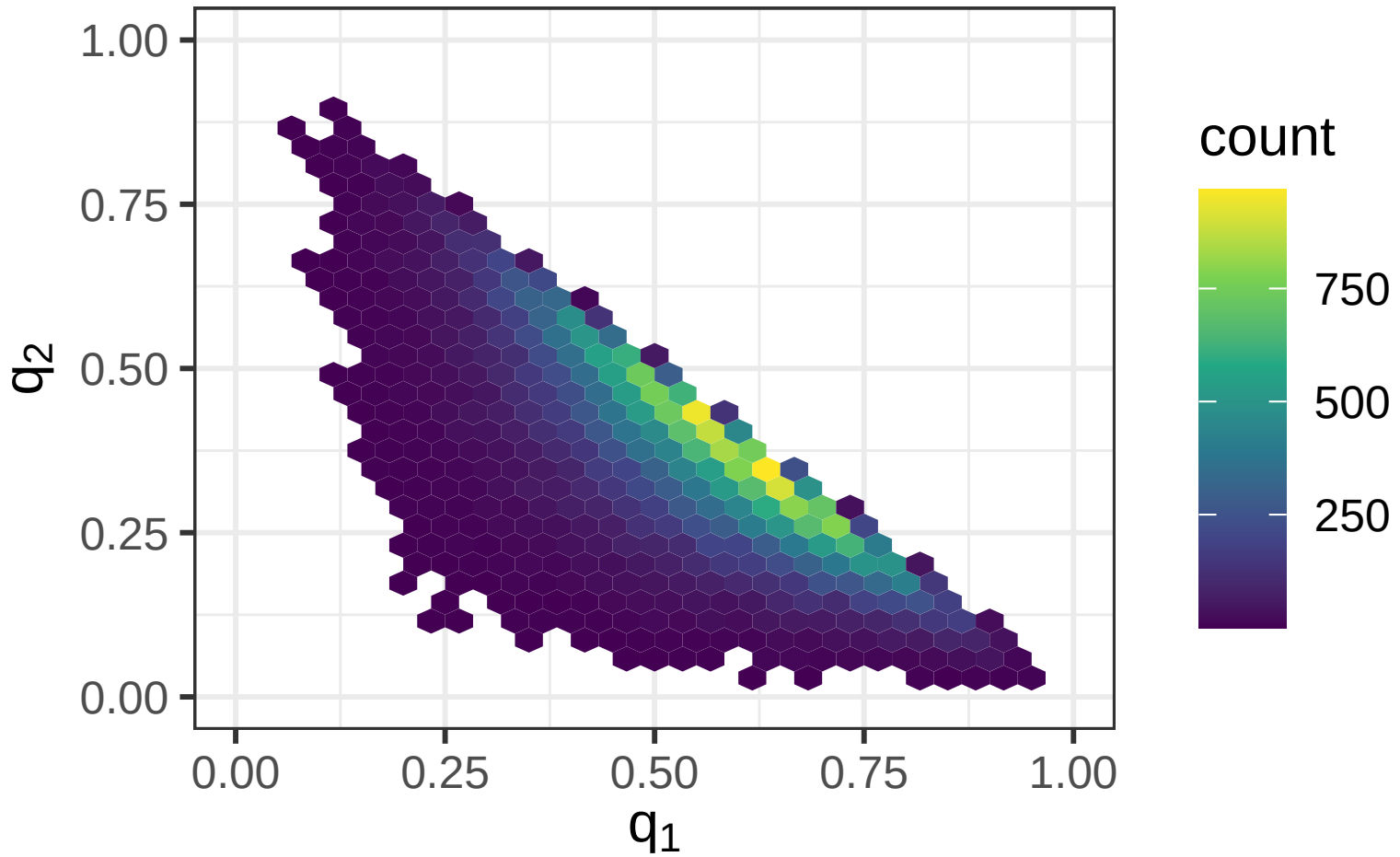
Displaying samples from a Dirichlet(5,5,5)

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



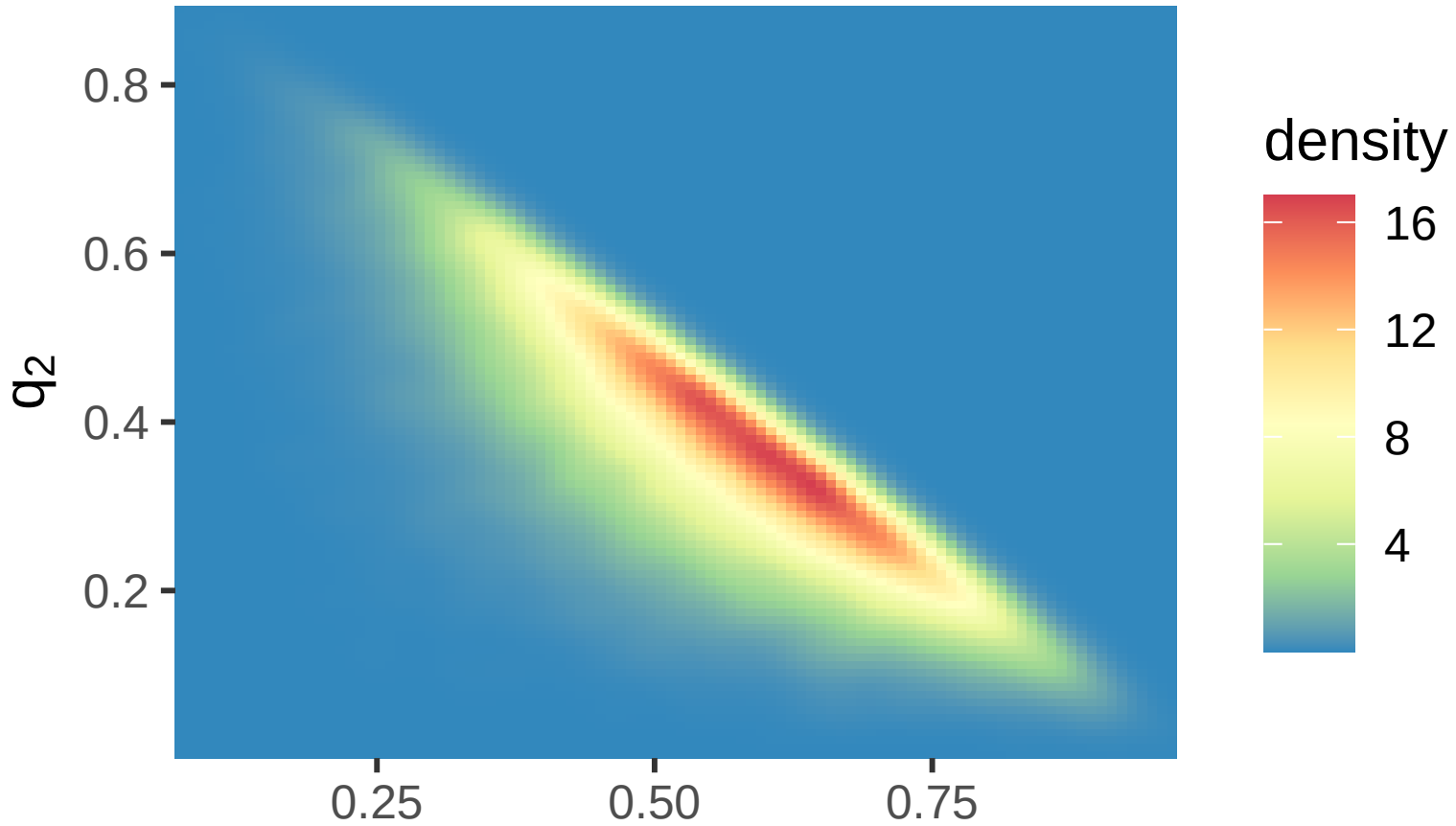
Displaying samples from a Dirichlet(6,4,1)

```
ggplot(data.frame(qex2[, 1], qex2[, 2]), aes(x = qex2[,
    1], y = qex2[, 2])) + geom_hex(bins = 30) + scale_fill_continuous(type = "viridi
    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

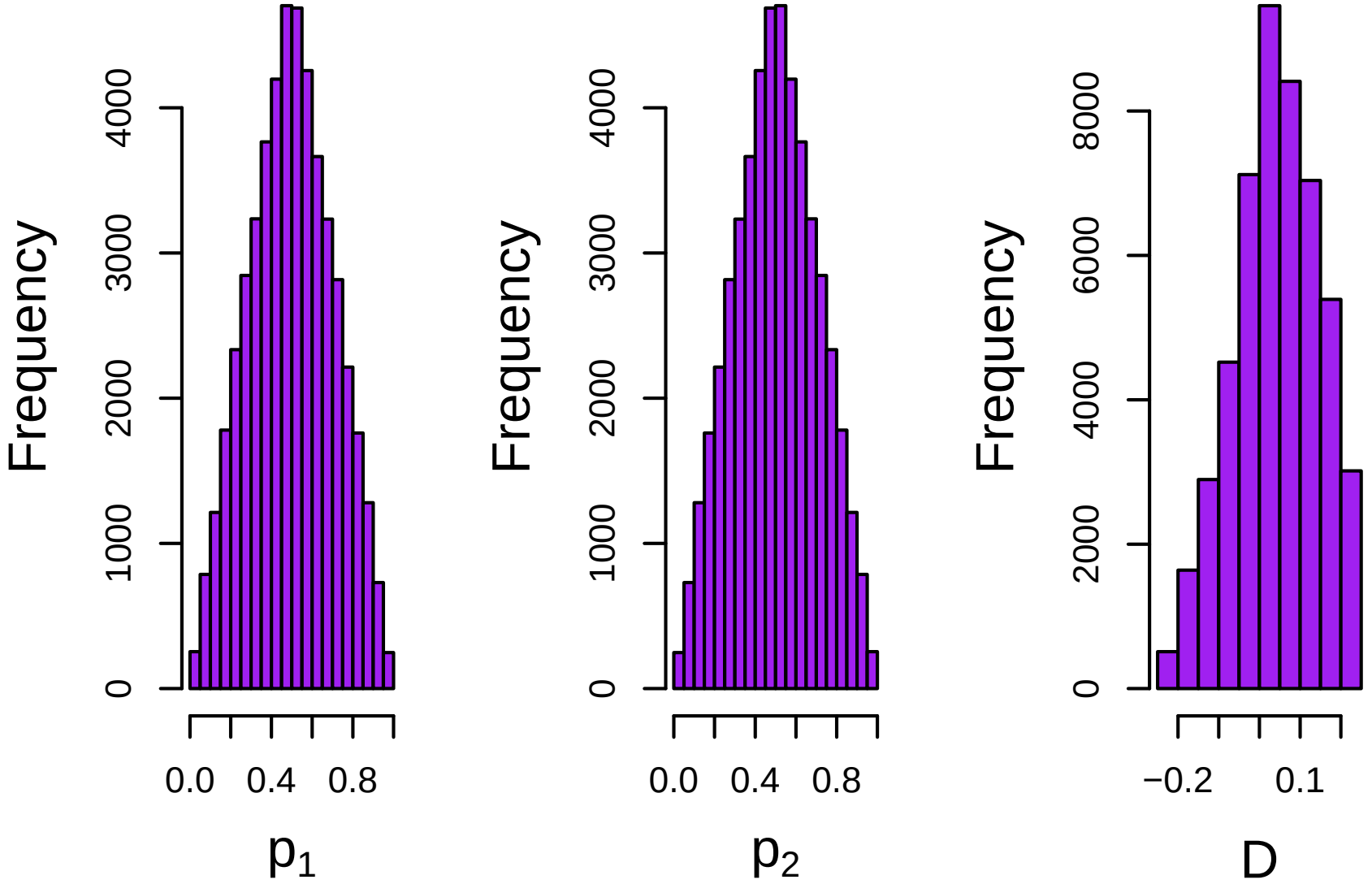
```
q1 <- q[, 1]
q2 <- q[, 2]
q3 <- q[, 3]
p1 <- q1 + q2/2
p2 <- q3 + q2/2
f <- (q1 - p1^2)/(p1 * p2)
D <- q1 - p1^2
psi <- q2^2/(p1 * p2)
## Functions of interest
cat("Prior prob f>0: ", sum(f > 0)/nsim, "\n")
## Prior prob f>0:  0.66628
cat("Prior prob D>0: ", sum(D > 0)/nsim, "\n")
## Prior prob D>0:  0.66628
```

Functions of interest

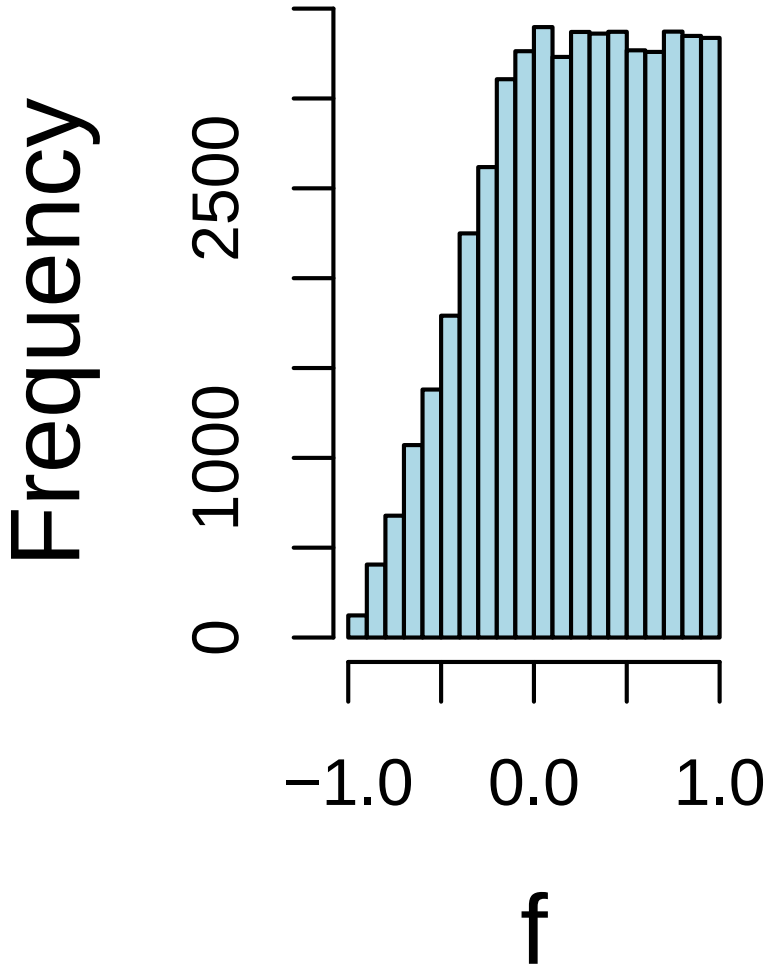
Examine prior summaries for different functions of interest.

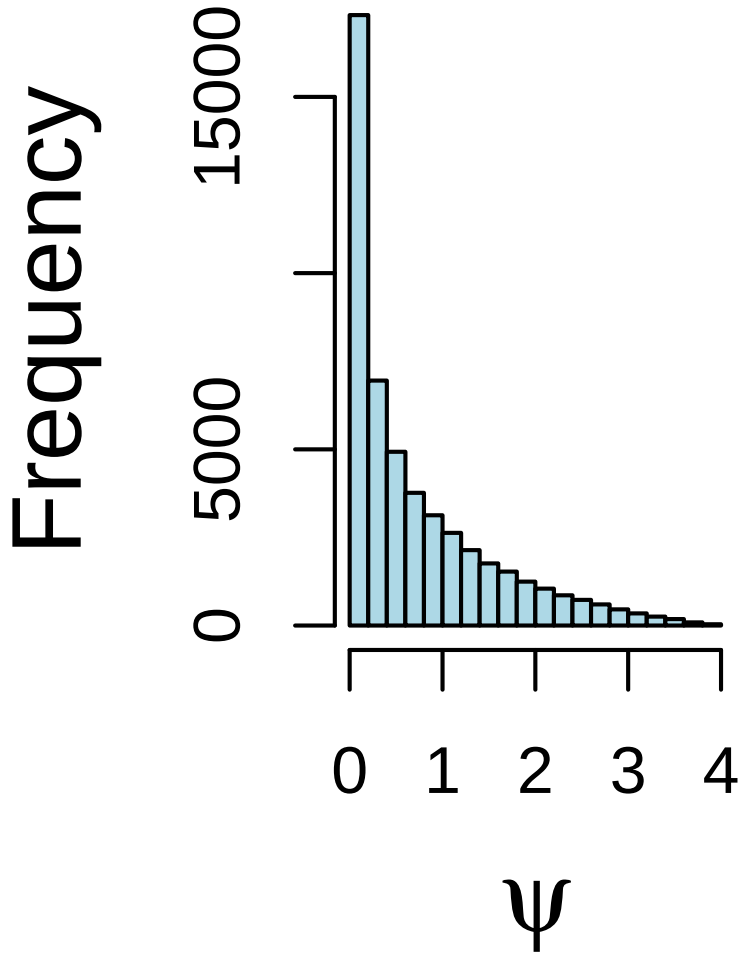
```
par(mfrow = c(1, 3))
hist(p1, main = "", xlab = expression(p[1]), cex.lab = 1.5)
hist(p2, main = "", xlab = expression(p[2]), cex.lab = 1.5)
hist(D, main = "", xlab = expression(D), cex.lab = 1.5)
par(mfrow = c(1, 2))
hist(f, main = "", xlab = "f", cex.lab = 1.5)
hist(psi, main = "", xlab = expression(psi),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

Functions of interest: priors on p_1, p_2, D

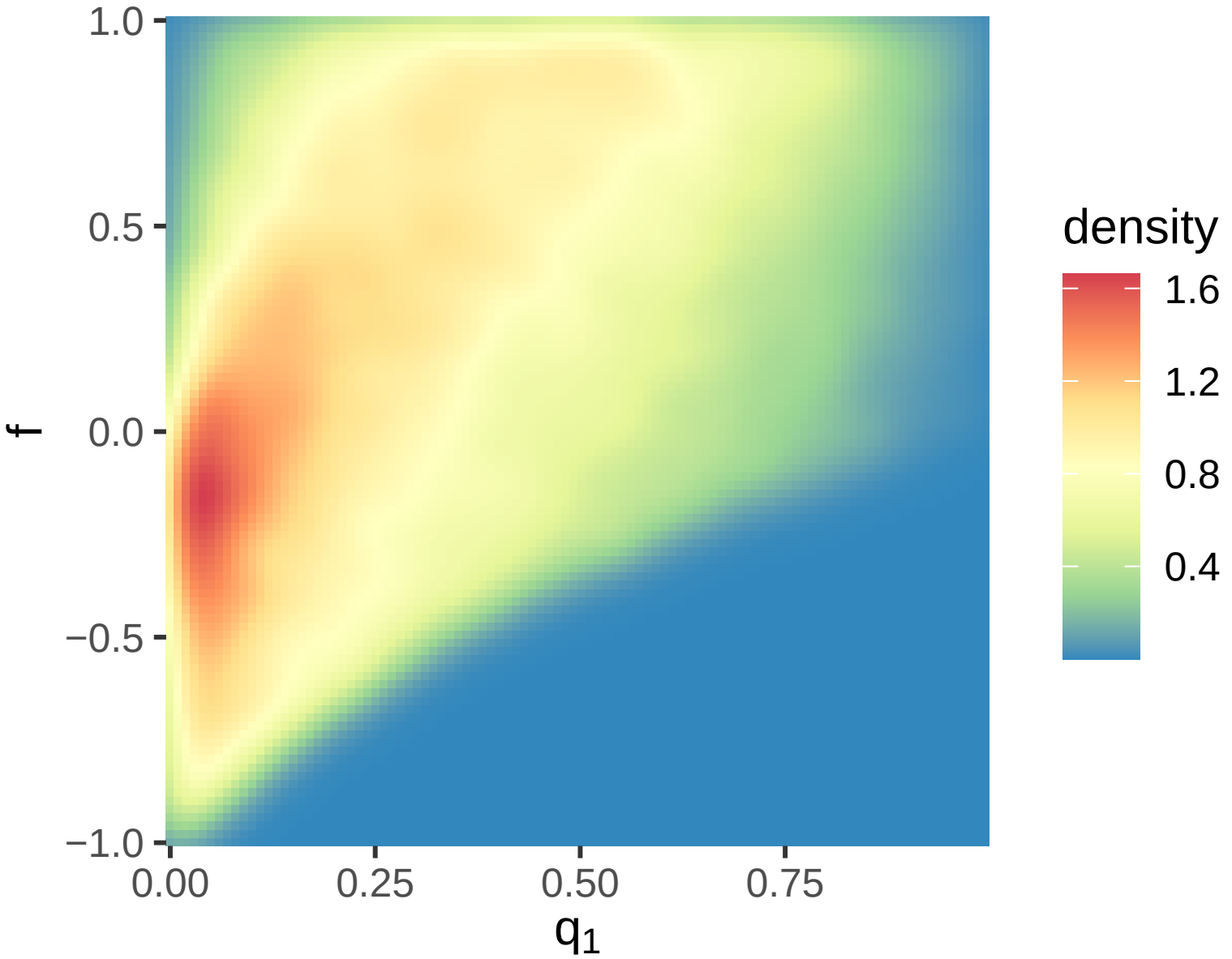


Functions of interest: priors on f, ψ.

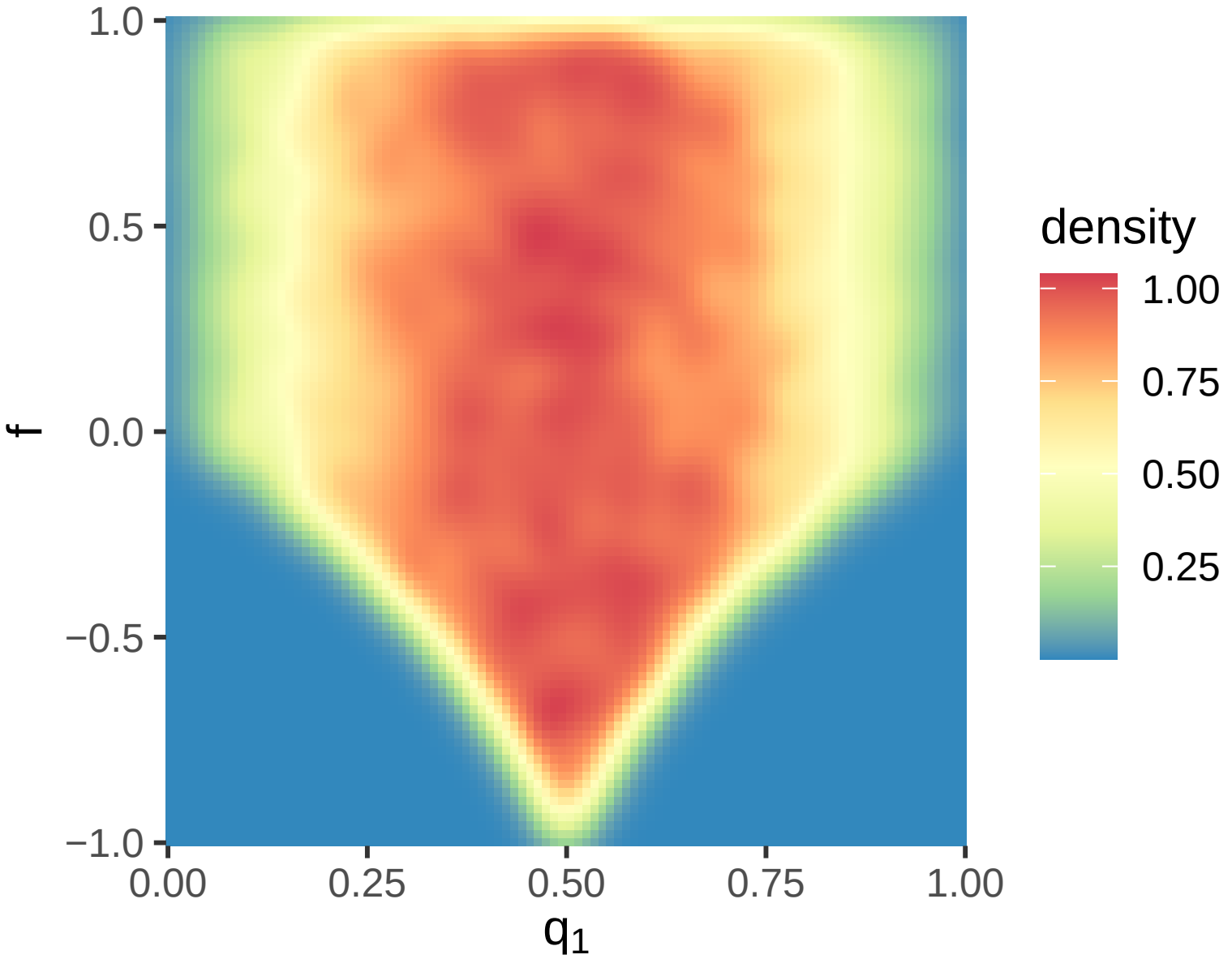




Prior on f and q_1



Prior on f and p_1

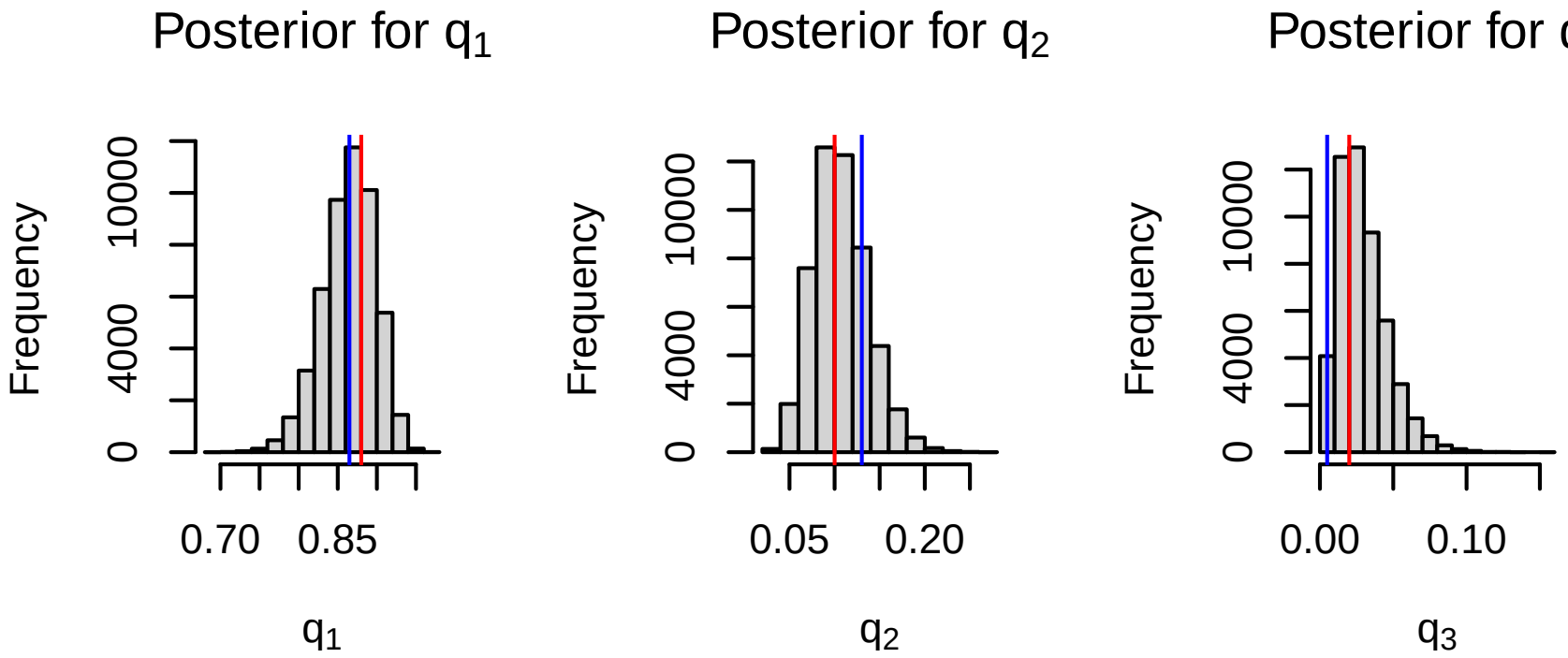


Bayes analysis of (88,10,2) data

```
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

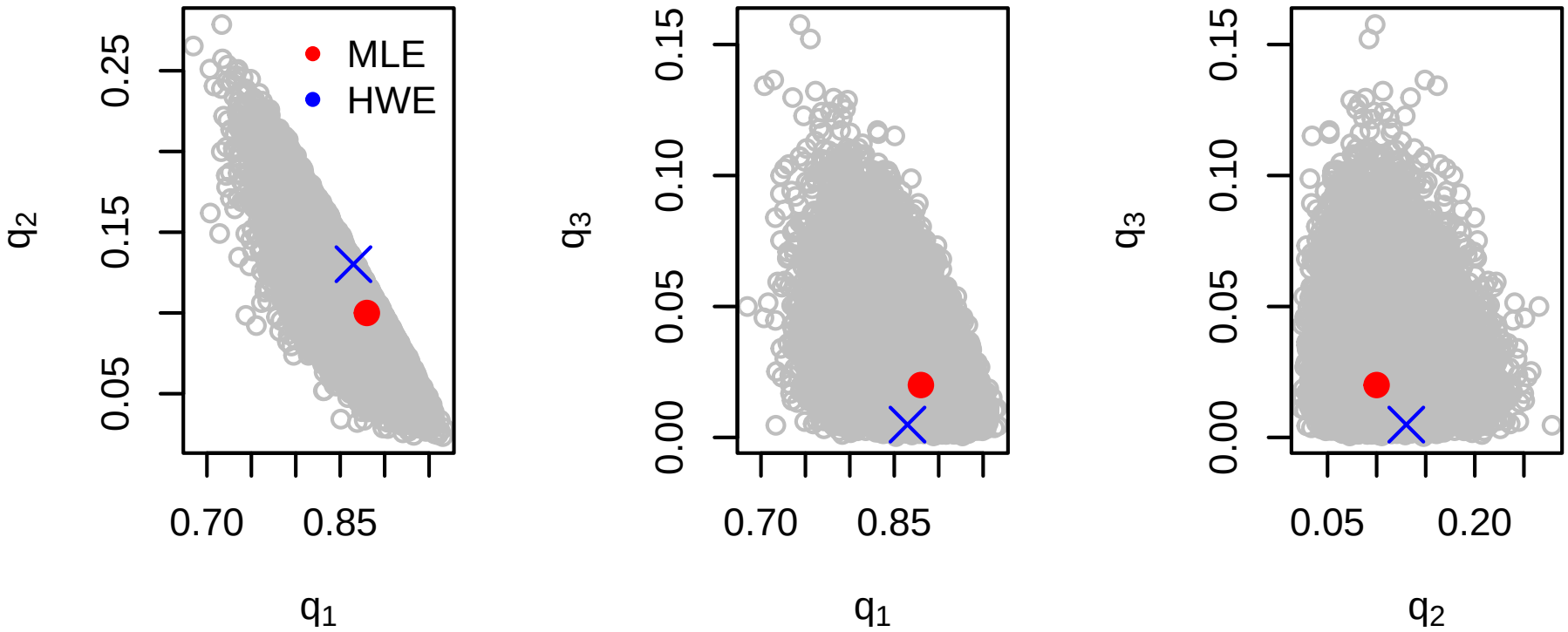


Bayes analysis of (88,10,2) data

```
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



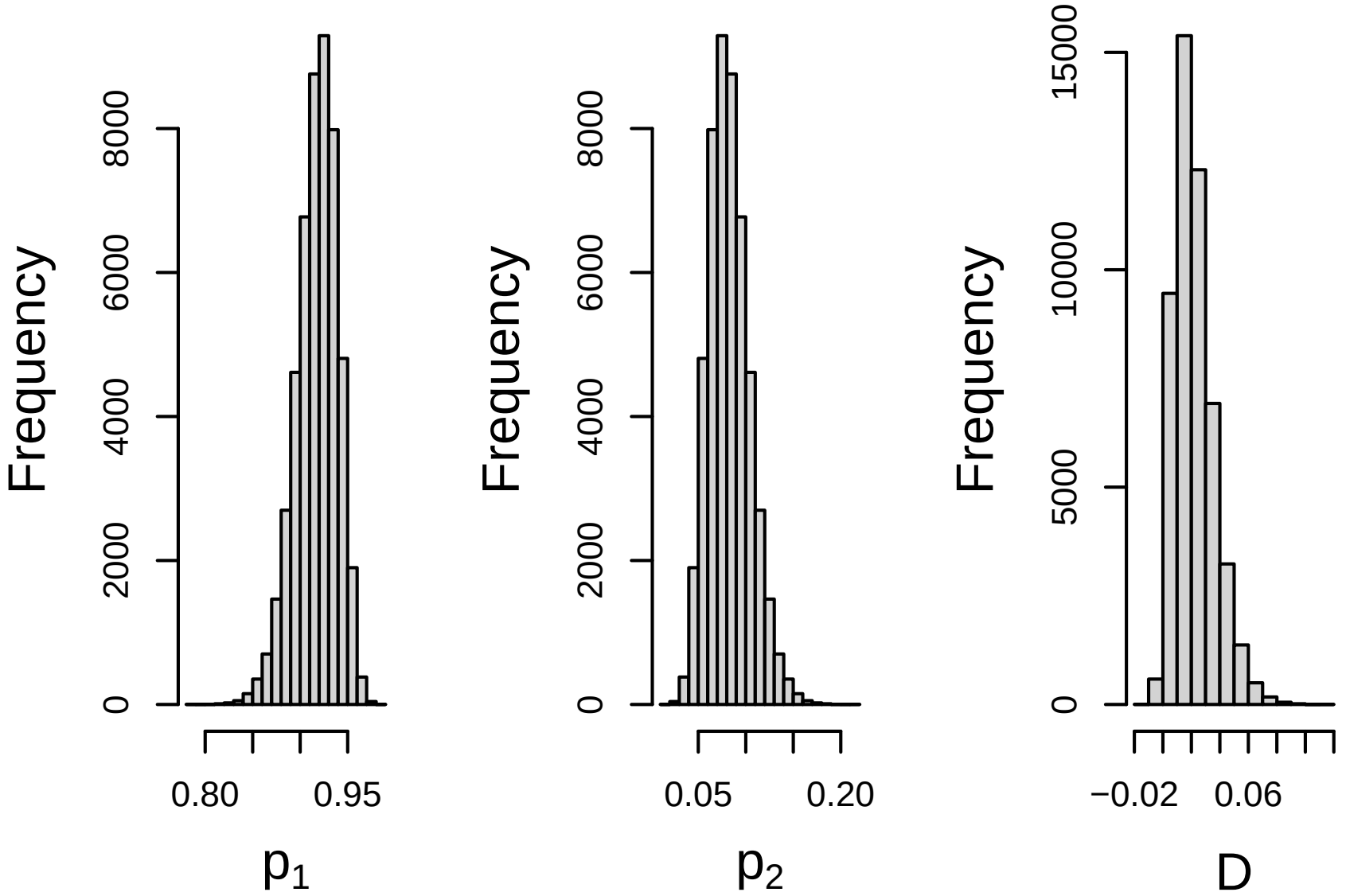
Inference for f

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

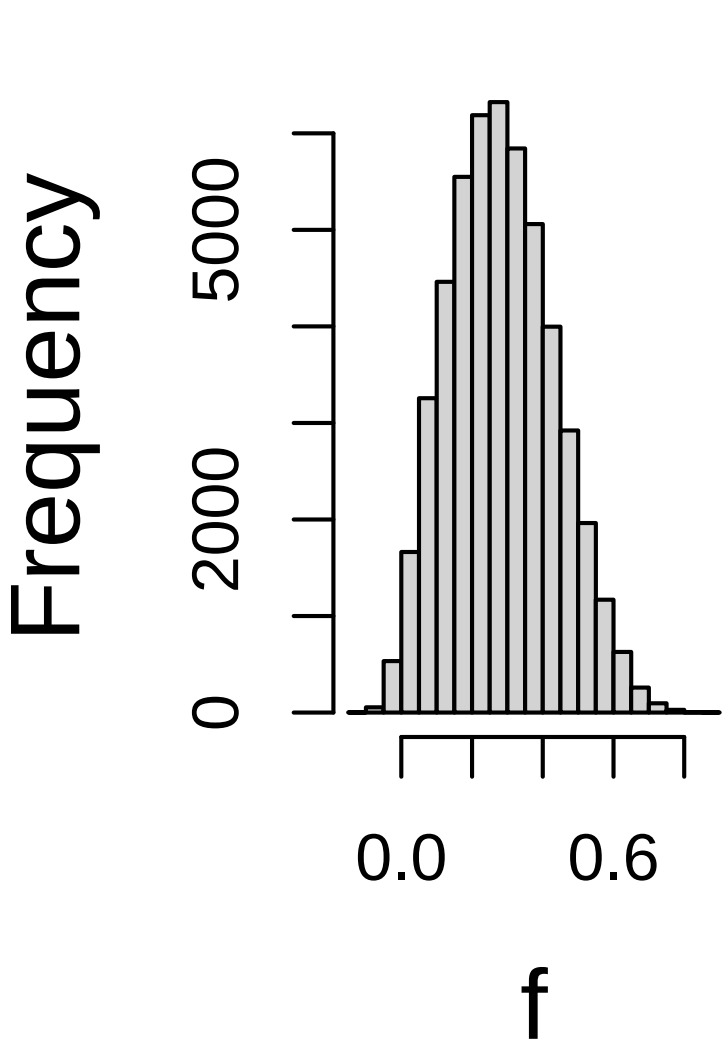
Hence, a 95% asymptotic confidence interval is

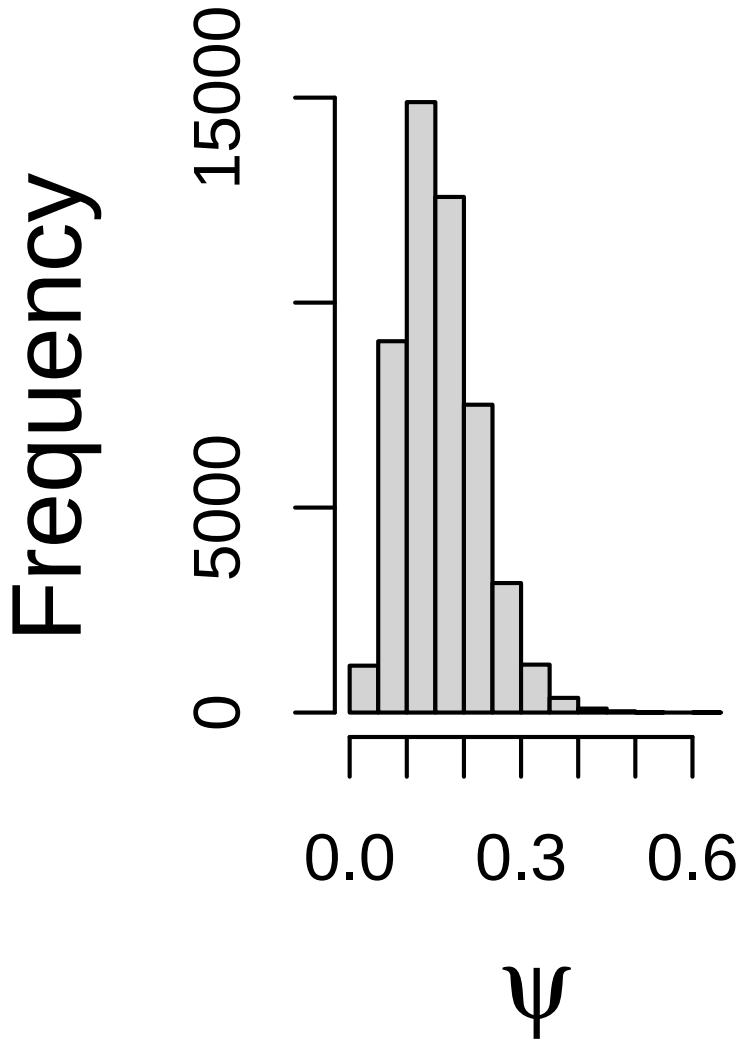
$$(0.23 - 1.96 \times 0.17, 0.23 + 1.96 \times 0.17) = (-0.1032, 0.5632).$$

Posterior summaries



Posterior summaries





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 $n_1 = 37$, $n_2 = 20$, $n_3 = 7$. 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