**8.1**

When *formula ~ group*, the p-value histograms for *x1* and *x2* match those in Figure 8.14. In *x1*, where batch effect is absent, the histogram shows a uniform distribution as shown in Figure 1. In *x2*, where batch effect is present, the histogram shows a depletion of small *p* values, as shown in Figure 2. This is expected since without the degree of freedom introduced by *batch*, the linear model will not be able to separate it from the contribution from *group*.

A picture containing screenshot, electronics, computer

Description automatically generated

Figure 1. *p* value histogram for *~ group* of *x1*

A screenshot of a cell phone

Description automatically generated

Figure 2. *p* value histogram for *~ group* of *x2*

In the case of *formula ~ batch + group*: In *x1*, where batch effect is absent, the histogram shows a uniform distribution as shown in Figure 3. In *x2*, where batch effect is present, the histogram shows a strong peak of small *p* values, as shown in Figure 4. Now that *batch* is part of the linear model, the *F* statistic is able to detect the batch effect.

*A screenshot of a computer

Description automatically generated*

Figure 3. *p* value histogram for *~ batch + group* of *x1*

A screenshot of a cell phone

Description automatically generated

Figure 4. *p* value histogram for *~ batch + group* of *x2*

In addition, for *formula ~ group*, the QQ plot in Figure 5 shows that the coefficients for *group* in both *x1* and *x2* share a nearly identical distribution.

A screenshot of a cell phone

Description automatically generated

Figure 5. QQ plot of *group* coefficients from *x1* and *x2*

R Code:

#8.1

#prepare the data matrix

library(ggplot2)

library(tibble)

library("magrittr")

ng = 10000

ns = 12

x1 = x2 = matrix(rnorm(ns \* ng), ncol = ns, nrow= ng)

group = factor(letters[1 + seq\_len(ns) %% 2]) %T>% print

batch = factor(ifelse(seq\_len(ns) <= ns/2, "B1", "B2")) %T>% print

x2[, batch=="B2"] = x2[, batch=="B2"] + 2 \* rnorm(ng)

# ~ group

model.matrix(~ group)

l <- c()

# extract the p-value for each row

extractP\_1 <- function(x){

for (i in seq(ng)){

batch\_lm <- lm(x[i,] ~ group)

f <- summary(batch\_lm)$fstatistic

p<- pf(f[1],f[2],f[3],lower.tail= FALSE)[["value"]]

l <- c(l,p)

}

return(l)

}

#process and plot set "x1"

P\_Group\_x1 <- extractP\_1(x1)

ggplot(tibble(p=P\_Group\_x1),aes(p))+geom\_histogram(binwidth = 0.02, boundary = 0)

#process and plot set "x2"

P\_Group\_x2 <- extractP\_1(x2)

ggplot(tibble(p=P\_Group\_x2),aes(p))+geom\_histogram(binwidth = 0.02, boundary = 0)

# ~ group + batch

model.matrix(~ group + batch)

l <- c()

# extract the p-value for each row

extractP\_2 <- function(x){

for (i in seq(ng)){

batch\_lm <- lm(x[i,] ~ group + batch)

f <- summary(batch\_lm)$fstatistic

p<- pf(f[1],f[2],f[3],lower.tail= FALSE)[["value"]]

l <- c(l,p)

}

return(l)

}

#process and plot set "x1"

P\_GroupBatch\_x1 <- extractP\_2(x1)

ggplot(tibble(p=P\_GroupBatch\_x1),aes(p))+geom\_histogram(binwidth = 0.02, boundary = 0)

#process and plot set "x2"

P\_GroupBatch\_x2 <- extractP\_2(x2)

ggplot(tibble(p=P\_GroupBatch\_x2),aes(p))+geom\_histogram(binwidth = 0.02, boundary = 0)

#group coefficents

l <- c()

extractCoefficient <- function(x){

for (i in seq(ng)){

batch\_lm <- lm(x[i,] ~ group)

C <- batch\_lm$coefficients[["groupb"]]

l <- c(l,C)

}

return(l)

}

GroupCoefficient\_x1 <- extractCoefficient(x1)

GroupCoefficient\_x2 <- extractCoefficient(x2)

qqplot(GroupCoefficient\_x1,GroupCoefficient\_x2)