# Fedorczyk_hm_lab4

load data

```
data(bladderdata)

# sample info
pheno = pData(bladderEset)
# expression data
edata = exprs(bladderEset)
row.variances <- apply(edata, 1, function(x) var(x))
edata <- edata[row.variances < 6,]
edata.log <- log2(edata)
```

*Homework Problem 1:* Create a table to show the batch effects (refer to Figure 1 in Gilad and Mizrahi-Man, 2015). There are 5 batches (`pheno$batch`); how are biological variables and other variables related to study design are distributed among those 5 batches? Explain what could be a problem. Prepare this into a PDF file.

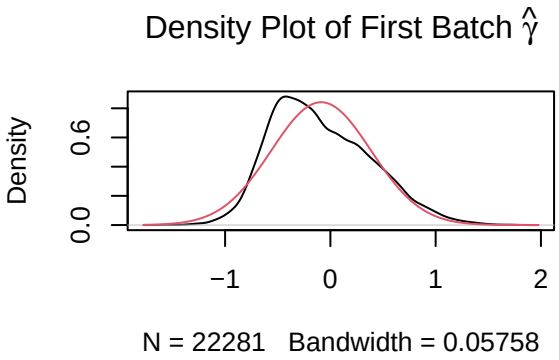
```
batch_table <- pheno[, c('batch', 'outcome', 'cancer')] %>% unique()

batch_table <- batch_table[order(batch_table$batch),] %>% as.data.table()

batch_table <- batch_table %>% group_by(batch)

batch_table <- batch_table %>% mutate(result = paste0('(', outcome, ', ', cancer, ')', collapse = ", ")

batch_table <- batch_table[, c('batch', 'result')] %>% unique()

result <- data.frame(batch_table$result %>% t)
colnames(result) <- paste0('batch', ' ', batch_table$batch)
rownames(result) <- '(outcome, cancer)'

pdf("Fedorczyk_problem1.pdf", height = 15, width = 15)
grid.table(result)
dev.off()
```

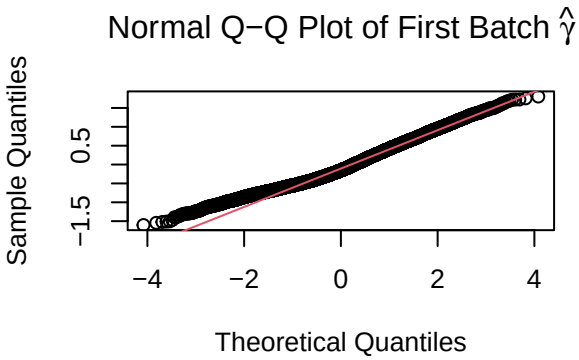
```
## pdf
##   2
```

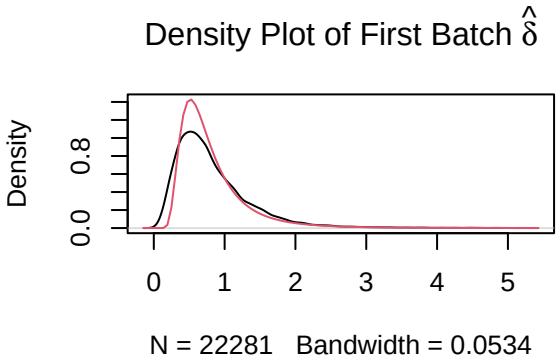
*Homework Problem 2:* Make heatmaps, BEFORE and AFTER cleaning the data using ComBat, where columns are arranged according to the study design. You must sort the columns such that 5 batches are shown. Cluster the rows, but do not cluster the columns (samples) when drawing a heatmap. The general idea is that you want to see if the Combat-cleaned data are any improvement in the general patterns.

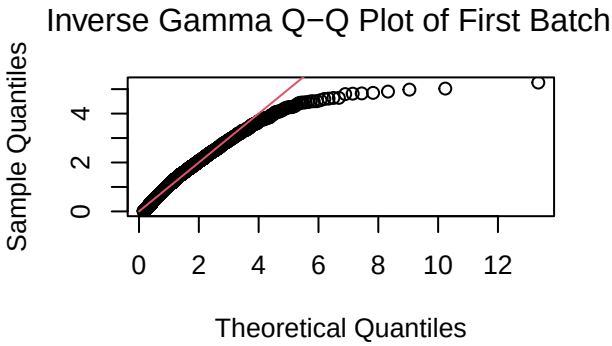
```
pheno_ordered <- pheno[order(pheno$batch, decreasing = FALSE),]
samples_ordered <- as.array(rownames(pheno_ordered))
edata_ordered <- edata[, samples_ordered]

combat_edata = ComBat(dat=edata, batch=pheno$batch, mod=model.matrix(~1, data=pheno), par.prior=TRUE, pr
```

```
## Found5batches

## Adjusting for0covariate(s) or covariate level(s)

## Standardizing Data across genes

## Fitting L/S model and finding priors

## Finding parametric adjustments

## Adjusting the Data
```




```
combat_edata_ordered <- combat_edata[, samples_ordered]

my_palette <- colorRampPalette(c("blue", "white", "darkred"))(n = 299)

# pdf("Fedorczyk_problem2_before.pdf", height = 10, width = 10)
# heatmap.2(edata_ordered,
#           main = "Bladder Cancer Data Clustered", # heat map title
```

```
#           notecol="black",      # change font color of cell labels to black
#           density.info="none",  # turns off density plot inside color legend
#           trace="none",         # turns off trace lines inside the heat map
#           margins =c(12,9),     # widens margins around plot
#           col=my_palette,       # use on color palette defined earlier
#           dendrogram="none",     # only draw a row dendrogram
#           scale = "row",
#           Colv = FALSE)
# dev.off()
#
# pdf("Fedorczyk_problem2_after.pdf", height = 10, width = 10)
# heatmap.2(combat_edata_ordered,
#           main = "Bladder Cancer Data Cleaned by ComBat", # heat map title
#           notecol="black",      # change font color of cell labels to black
#           density.info="none",  # turns off density plot inside color legend
#           trace="none",         # turns off trace lines inside the heat map
#           margins =c(12,9),     # widens margins around plot
#           col=my_palette,       # use on color palette defined earlier
#           dendrogram="none",     # only draw a row dendrogram
#           scale = "row",
#           Colv = FALSE)
# dev.off()
```

*Homework Problem 3:* Make heatmaps of Pearson correlations statistics of samples. For example, see Figure 2 and 3 freom Gilad and Mizrahi-Man (2015) F1000Research: https://f1000research.com/articles/4-121. First, compute the correlation statistics among columns. Second, create a heatmap using heatmap.2(). Make sure to create or add labels for samples (cancer vs. normal; batch numbers; others)

```
correlations <- cor(edata)

labels <- paste0(pheno$cancer, " b", pheno$batch)

pdf("Fedorczyk_problem3.pdf", height = 9, width = 9)
heatmap.2(correlations,
          main = "Pearson correlations statistics of samples.",
          notecol="black",
          density.info="none",
          trace="none",
          margins =c(12,9),
          col=my_palette,
          dendrogram="none",
          scale = "row",
          labRow=labels,
          labCol=labels)
dev.off()
```

```
## pdf
##   2
```

*Homework Problem 4:* Apply two different Linear Models to the Bottomly et al. data. First, using a conventional approach, create a linear model with a genetic strain (biological variable)

and an experimental number (technical variable) on **uncorrected** gene expression data. Second, create a linear model with a genetic strain (biological variables) on **corrected** gene expression data from ComBat. Make a scatter plots of coefficients and a histogram of p-values as done in this notebook. Make sure that you are pulling out the correct coefficients, not any or all coefficients.

```
con <- url("http://bowtie-bio.sourceforge.net/recount/ExpressionSets/bottomly_eset.RData")
load(file=con)
close(con)
save(bottomly.eset, file="bottomly.Rdata")

load(file="bottomly.Rdata")

pheno <- pData(bottomly.eset)
edata <- exprs(bottomly.eset)
edata <- edata[rowMeans(edata) > 10, ]
edata <- log2(as.matrix(edata) + 1)
```

## ComBat

```
combat_edata = ComBat(dat=edata, batch=pheno$experiment.number, mod=model.matrix(~1, data=pheno), par.p
```

```
## Found3batches

## Adjusting for0covariate(s) or covariate level(s)

## Standardizing Data across genes

## Fitting L/S model and finding priors

## Finding parametric adjustments

## Adjusting the Data
```

```
#Model 1
mod1 = lm(t(edata) ~ as.factor(pheno$strain) + as.factor(pheno$experiment.number))
mod1_tidy <- tidy(mod1)

#histogram
pdf('Fedorczyk_problem4_p_value_model1.pdf')
ggplot(mod1_tidy %>% filter(term == "as.factor(pheno$strain)DBA/2J")) + geom_histogram(aes(x=p.value), 
dev.off()
```

```
## pdf
##   2
```

```
#Model 2
mod2 = lm(t(combat_edata) ~ as.factor(pheno$strain))
mod2_tidy <- tidy(mod2)

#histogram
pdf('Fedorczyk_problem4_p_value_model2.pdf')
ggplot(mod2_tidy %>% filter(term == 'as.factor(pheno$strain)DBA/2J')) + geom_histogram(aes(x=p.value), l
dev.off()
```

```
## pdf
##   2
```

```
#Comparison

#filter : choose ROWS
#select : choose COLS

est_compare <- tibble(
  LinearModel = mod1_tidy %>% filter(term == "as.factor(pheno$strain)DBA/2J") %>% select("estimate") %>%
  ComBat = mod2_tidy %>% filter(term == "as.factor(pheno$strain)DBA/2J") %>% select("estimate") %>% unli
pdf('Fedorczyk_problem4_compare.pdf')
ggplot(est_compare, aes(x=LinearModel, y=ComBat)) +
     geom_point(col="darkgrey", alpha=.5, size=.5) + geom_abline(intercept=0, slope=1, col="darkred") +
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
dev.off()
```

```
## pdf
##   2
```

*Homework Problem 5:* Apply ComBat and SVA to the Bottomly et al. data. Make a scatter plots of coefficients and a histogram of p-values, comparing results based on ComBat and SVA. Assume that the biological variables in Bottomly et al data is the genetic strains. Make sure that you are pulling out the correct coefficients/pvalues, not any or all of them.

```
con <- url("http://bowtie-bio.sourceforge.net/recount/ExpressionSets/bottomly_eset.RData")
load(file=con)
close(con)
save(bottomly.eset, file="bottomly.Rdata")

load(file="bottomly.Rdata")

pheno <- pData(bottomly.eset)
edata <- exprs(bottomly.eset)
edata <- edata[rowMeans(edata) > 10, ]
edata <- log2(as.matrix(edata) + 1)
```

## ComBat

```
combat_edata = ComBat(dat=edata, batch=pheno$experiment.number, mod=model.matrix(~1, data=pheno), par.p
```

```
## Found3batches
```

```
## Adjusting for0covariate(s) or covariate level(s)
```

```
## Standardizing Data across genes
```

```
## Fitting L/S model and finding priors
```

```
## Finding parametric adjustments
```

```
## Adjusting the Data
```

```
modcombat = lm(t(combat_edata) ~ as.factor(pheno$strain))

modcombat_tidy <- tidy(modcombat)
```

### sva

```
modsva = model.matrix(~as.factor(strain),data=pheno)
modsva0 = model.matrix(~1, data=pheno)

sva_output = sva::sva(edata, modsva, modsva0, n.sv=sva::num.sv(edata,modsva,method="leek"))
```

```
## Number of significant surrogate variables is:  1
## Iteration (out of 5 ):1  2  3  4  5
```

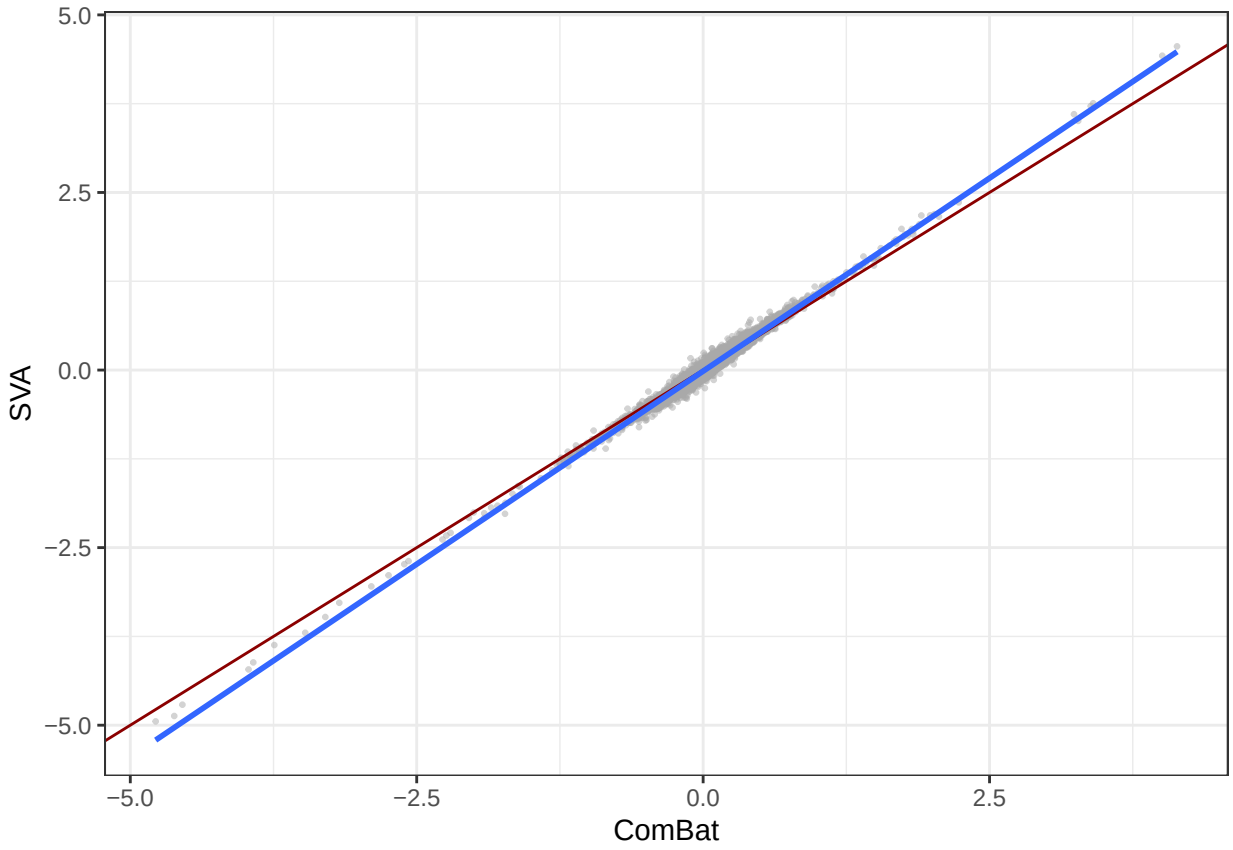
```
modsva = lm(t(edata) ~ as.factor(pheno$strain) + sva_output$sv)
modsva_tidy <- tidy(modsva)
```

### comparison

#Coefficients

```
est_compare <- tibble(

  ComBat = modcombat_tidy %>% filter(term == "as.factor(pheno$strain)DBA/2J") %>% select("estimate") %>%

  SVA = modsva_tidy %>% filter(term == "as.factor(pheno$strain)DBA/2J") %>% select("estimate") %>% unli

ggplot(est_compare, aes(x=ComBat, y=SVA)) +
  geom_point(col="darkgrey", alpha=.5, size=.5) + geom_abline(intercept=0, slope=1, col="darkred") + ge
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
#save to pdf
pdf("Fedorczyk_problem5_coef.pdf")
ggplot(est_compare, aes(x=ComBat, y=SVA)) +
  geom_point(col="darkgrey", alpha=.5, size=.5) + geom_abline(intercept=0, slope=1, col="darkred") + ge
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
dev.off()
```

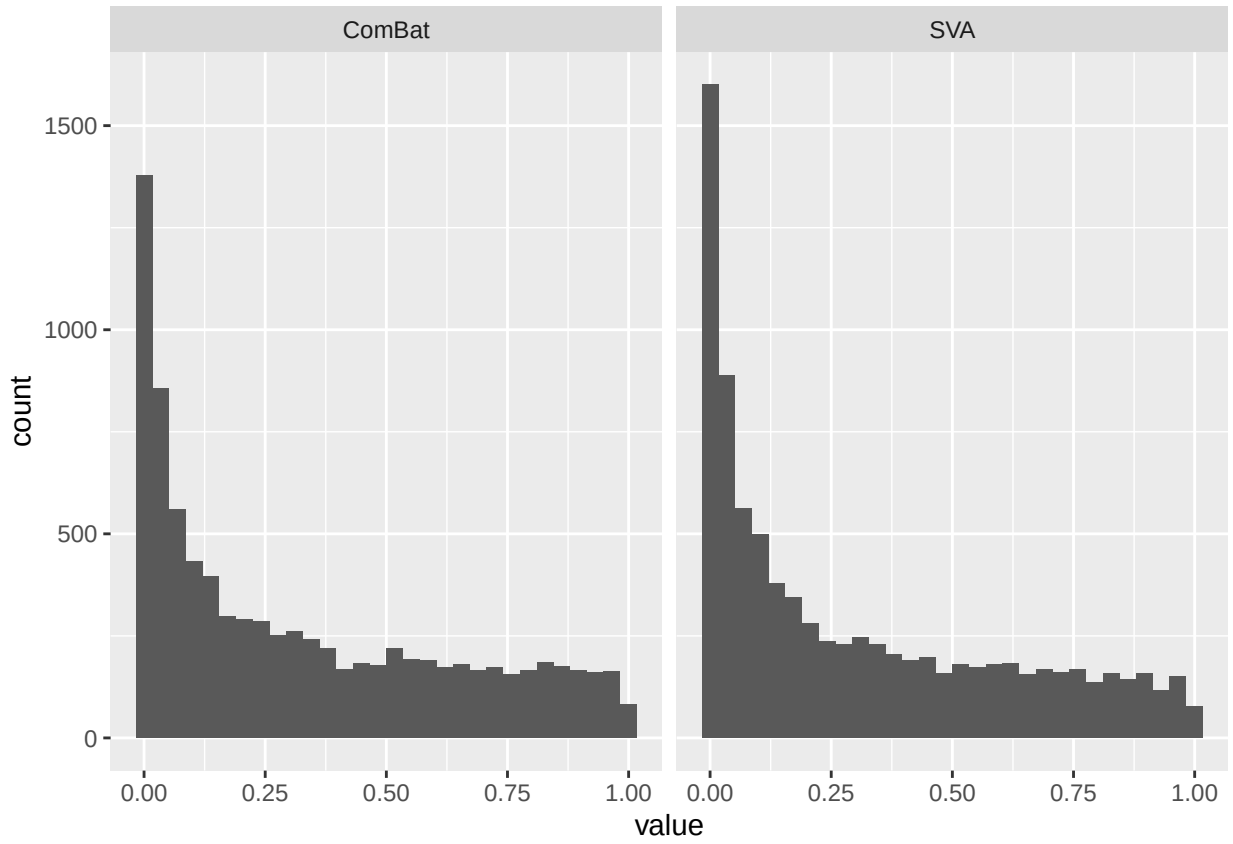
```
## pdf
##   2
```

```
# ggsave('problem5_coef.png')
```

#pvalues

```
pvalues <- tibble(
  ComBat = modcombat_tidy %>% filter(term == "as.factor(pheno$strain)DBA/2J") %>% select("p.value") %>%
  SVA = modsva_tidy %>% filter(term == "as.factor(pheno$strain)DBA/2J") %>% select("p.value") %>% unlis

pvalues_gather <- gather(pvalues)
ggplot(pvalues_gather, aes(x=value)) + geom_histogram() + facet_wrap(~key)
```

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




```
#save to pdf
pdf("Fedorczyk_problem5_pvalue.pdf")
ggplot(pvalues_gather, aes(x=value)) + geom_histogram() + facet_wrap(~key)
```

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

```
dev.off()
```

```
## pdf
##   2
```

```
# ggsave('problem5_pvalue.png')
```