Expression Level and Measurement Precision

Lei Sun

July 13, 2016

In RNA-seq analysis, it is assumed that highly expressed genes would have high measurement precision, thus high power. Now we are using GTEx data sets to explore whether it's the case in real data.

```
set.seed(1)
```

Null data, generated by Poisson only

We simulate data such as $c_{ij} \sim Poisson(\mu_i)$, with $\mu_i = 1, 10, 100, 1000$.

```
Nsample = 5
counts = c(rpois(1000 * 2 * Nsample, 1), rpois(1000 * 2 * Nsample, 10), rpois(1000 * 2 * Nsample, 100),
counts = matrix(counts, ncol = 2 * Nsample, byrow = TRUE)
condition = rep(1:2, each = Nsample)
```

With the simulated count matrix and the condition vector, we could get a $\hat{\beta}$ and $\hat{s}$, as well as a z-score $= \hat{\beta}/\hat{s}$ for each gene by both OLS and voom + limma pipeline.

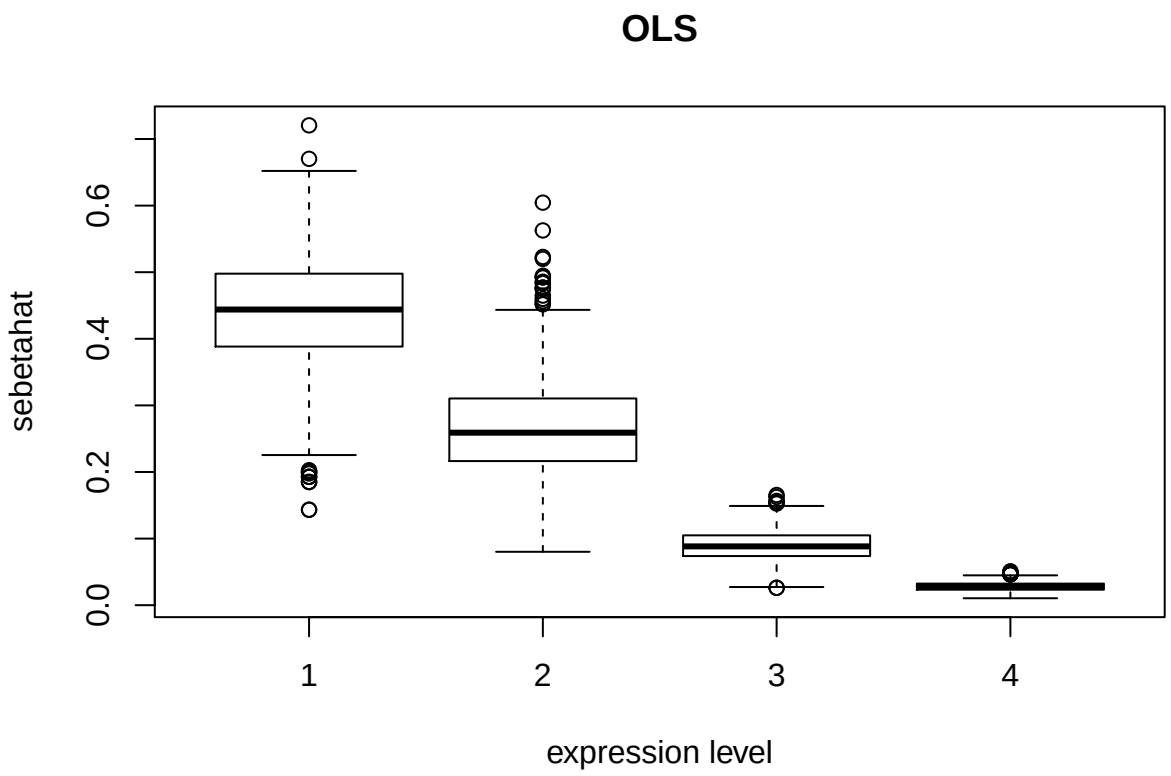
```
source("../code/fit_method.R")
```

```
## Loading required package: limma
```

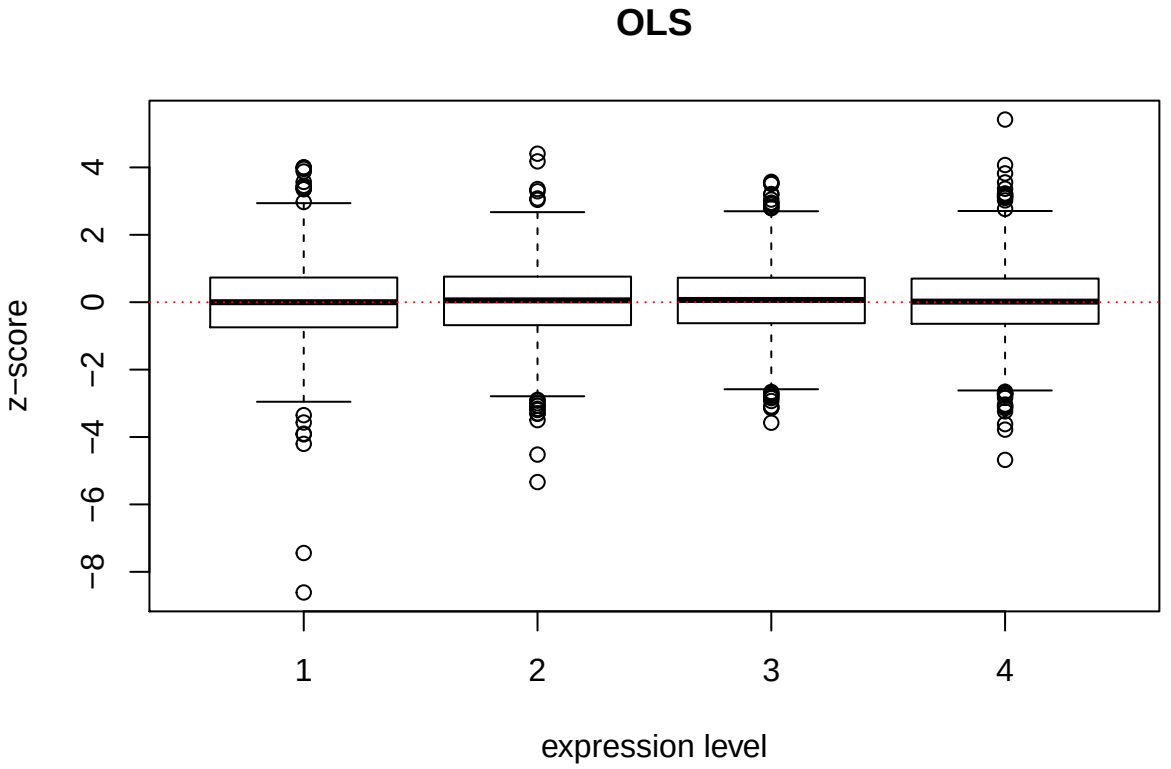
```
    ## effect size and standard error estimated by OLS
    log_counts = log2(counts + 1)
    ols_fit <- get_ols(log_counts = log_counts, condition = condition)
    betahat_ols = ols_fit$betahat
    sebetahat_ols = ols_fit$sebetahat
    df_ols = ols_fit$df
    z_ols = betahat_ols / sebetahat_ols

    ## effect size and measurement error estimated by voom + limma
    voom_fit = voom_transform(counts, condition)
    betahat_voom = voom_fit$betahat
    sebetahat_voom = voom_fit$sebetahat
    df_voom = voom_fit$df
    z_voom = betahat_voom / sebetahat_voom
```

We group the genes to 4 bins of 1000 genes. Then we plot $\hat{s}$ and z-score with respect to the expression level.

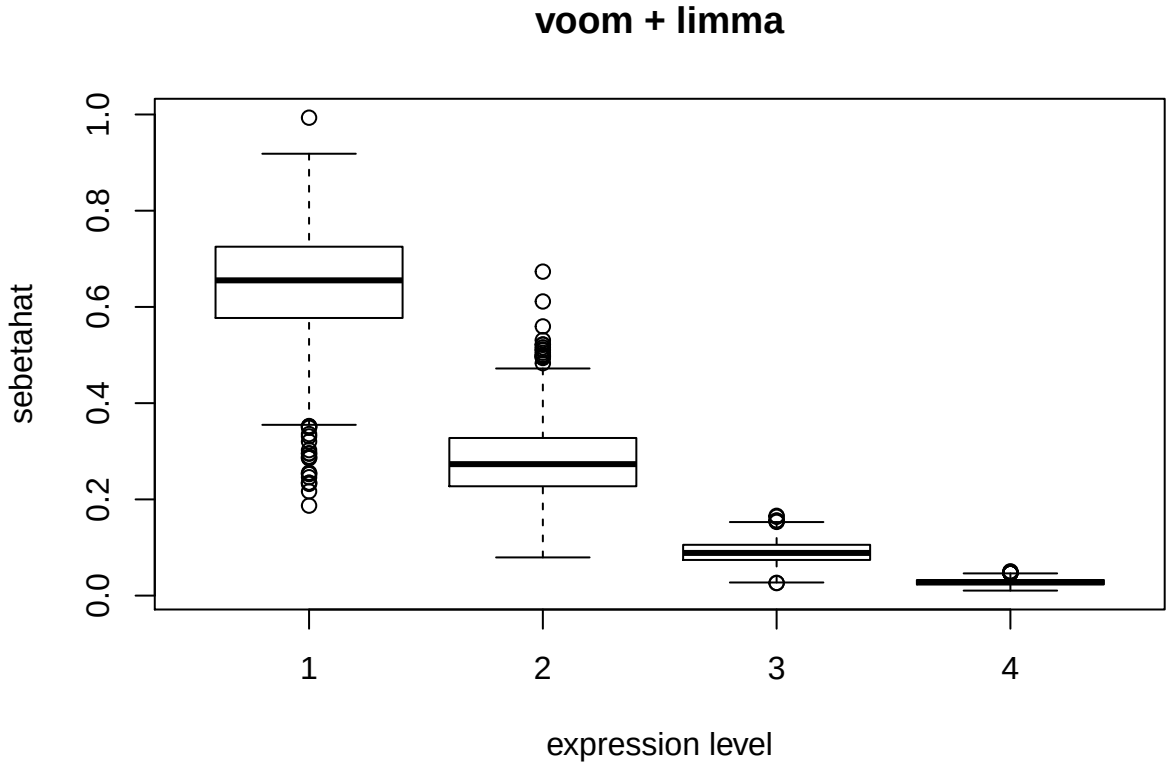
```
        ## grouping genes into K subgroups from low to high expression
  group_id = rep(1:4, each = 1000)

  boxplot(sebetahat_ols ~ group_id, xlab = "expression level", ylab = "sebetahat", main = "OLS")
```



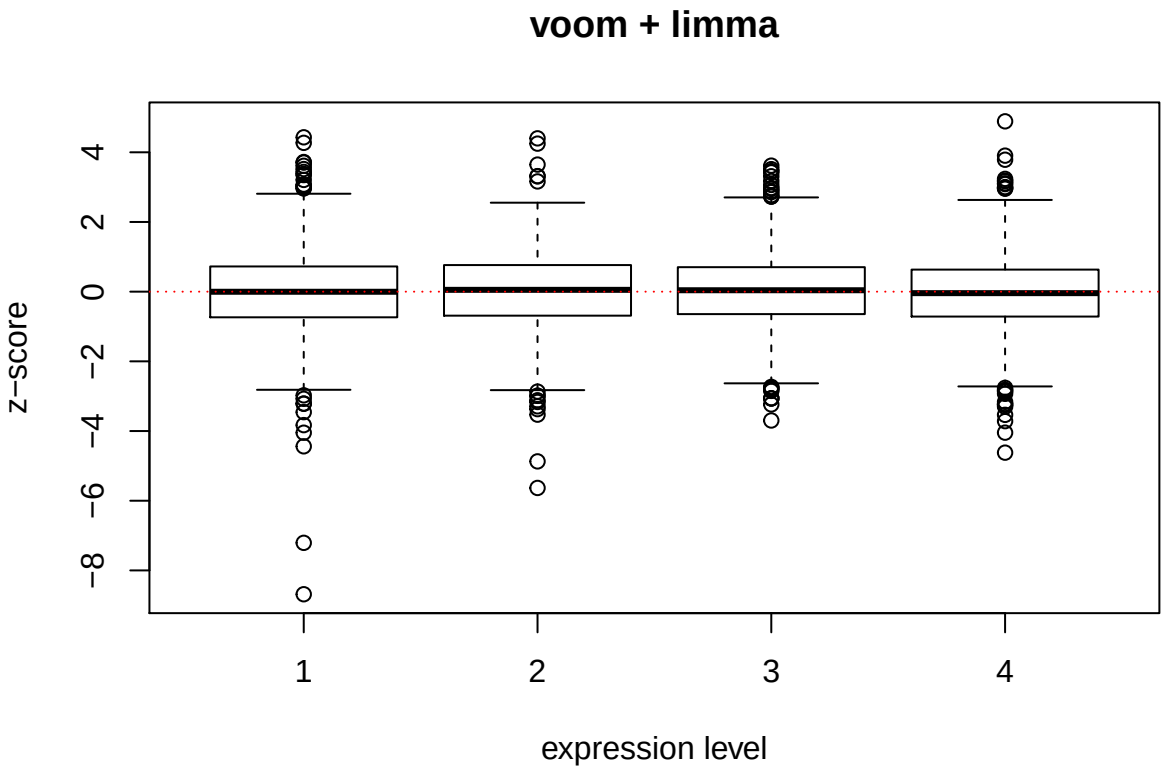

```
boxplot(z_ols ~ group_id, xlab = "expression level", ylab = "z-score", main = "OLS")
abline(0, 0, lty = 3, col = "red")
```



```r
boxplot(sebetahat_voom ~ group_id, xlab = "expression level", ylab = "sebetahat", main = "voom + limma
```




```r
boxplot(z_voom ~ group_id, xlab = "expression level", ylab = "z-score", main = "voom + limma")
abline(0, 0, lty = 3, col = "red")
```



Alternative, generated by Poisson only

We simulate data such as $c_{ij} \sim Poisson(\mu_i)$, with $\mu_i = 1, 10, 100, 1000$ as control, $\mu_i = 2, 20, 200, 2000$ as control.

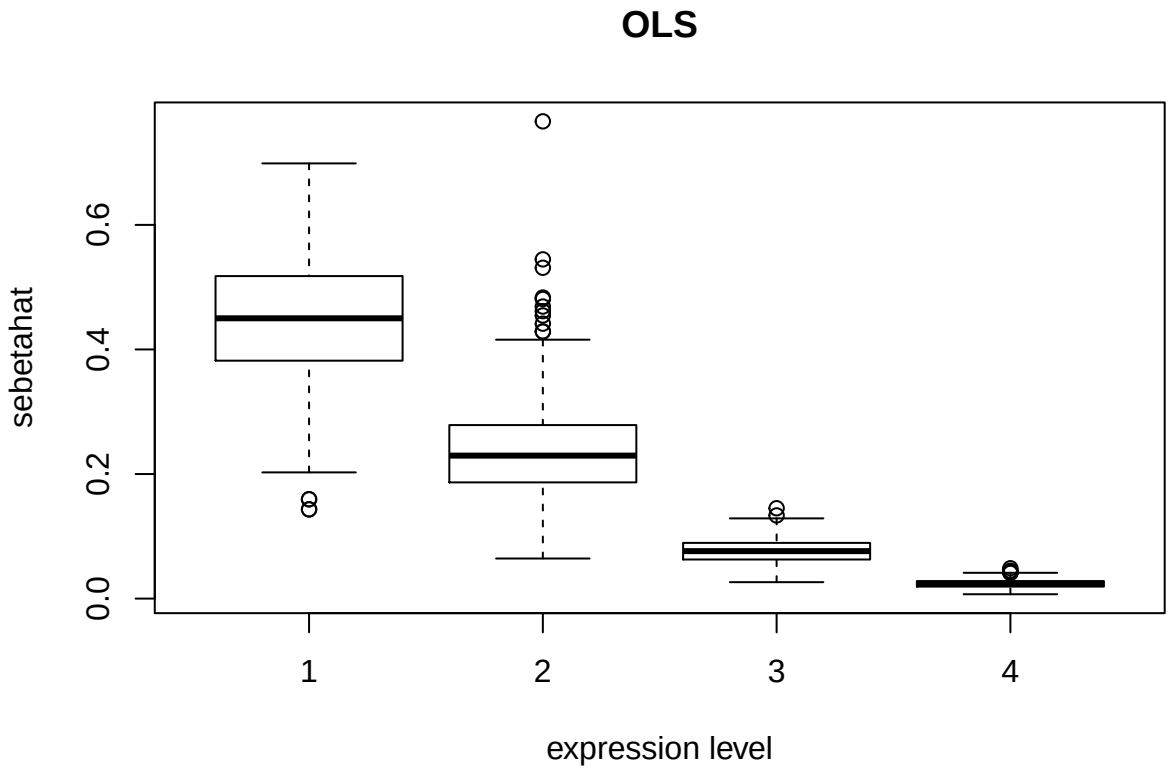
```
Nsample = 5
b = 2
counts_case = c(rpois(1000 * Nsample, 1), rpois(1000 * Nsample, 10), rpois(1000 * Nsample, 100), rpois(
counts_case = matrix(counts_case, ncol = Nsample, byrow = TRUE)
counts_control = c(rpois(1000 * Nsample, 1 * b), rpois(1000 * Nsample, 10 * b), rpois(1000 * Nsample, 1
counts_control = matrix(counts_control, ncol = Nsample, byrow = TRUE)
counts = cbind(counts_case, counts_control)
condition = rep(1:2, each = Nsample)
```

With the simulated count matrix and the condition vector, we could get a $\hat{\beta}$ and $\hat{s}$, as well as a z-score $= \hat{\beta}/\hat{s}$ for each gene by both OLS and voom + limma pipeline.

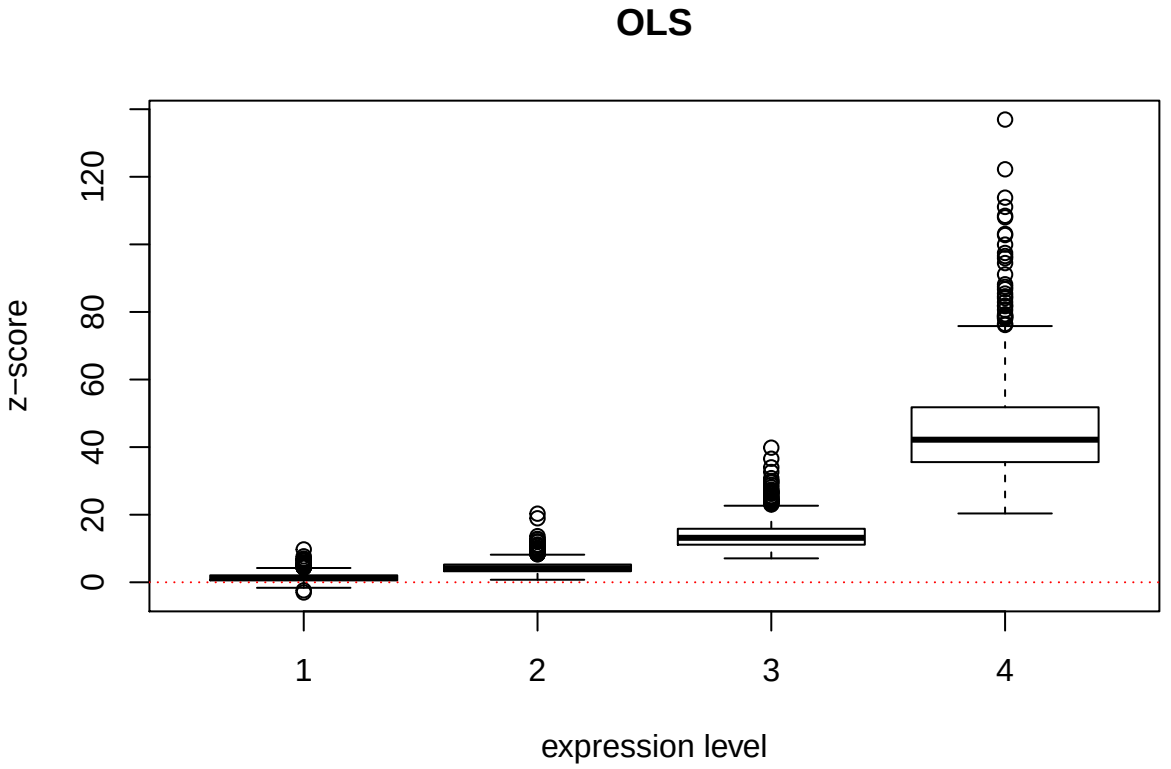
```
source("../code/fit_method.R")

    ## effect size and standard error estimated by OLS
    log_counts = log2(counts + 1)
    ols_fit <- get_ols(log_counts = log_counts, condition = condition)
    betahat_ols = ols_fit$betahat
    sebetahat_ols = ols_fit$sebetahat
    df_ols = ols_fit$df
    z_ols = betahat_ols / sebetahat_ols

    ## effect size and measurement error estimated by voom + limma
    voom_fit = voom_transform(counts, condition)
    betahat_voom = voom_fit$betahat
    sebetahat_voom = voom_fit$sebetahat
    df_voom = voom_fit$df
    z_voom = betahat_voom / sebetahat_voom
```

We group the genes to 4 bins of 1000 genes. Then we plot $\hat{s}$ and z-score with respect to the expression level.

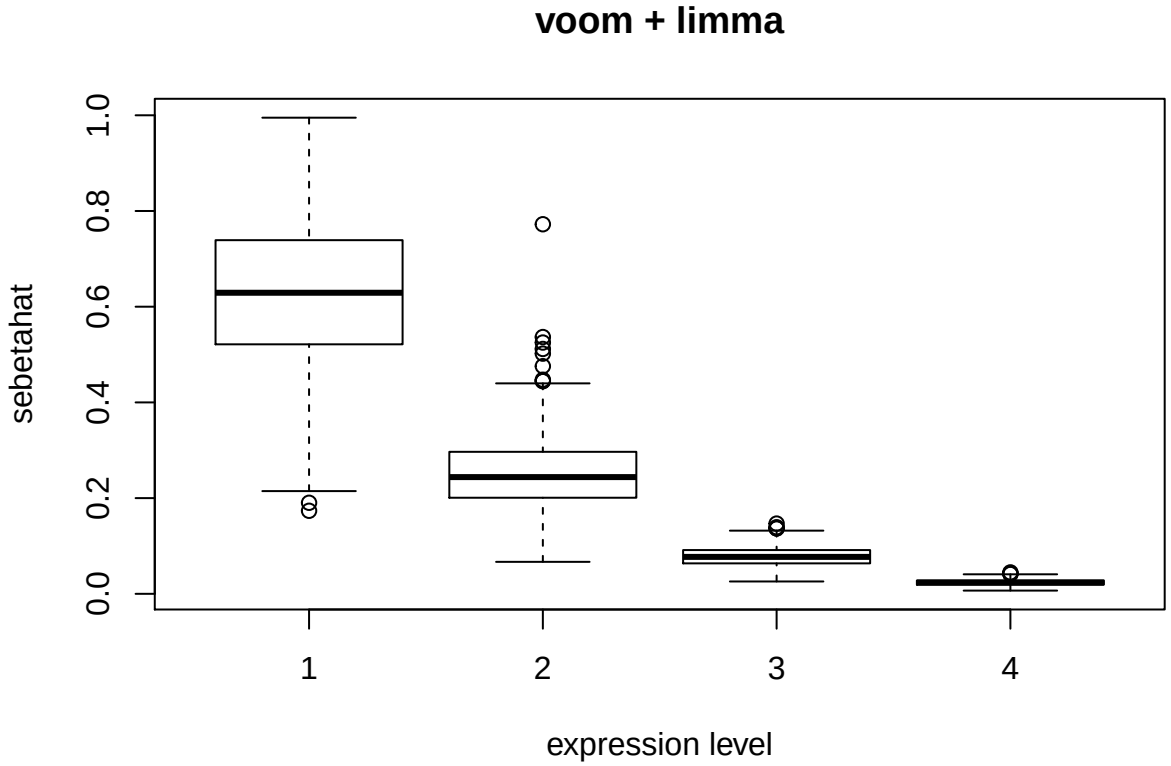
```
        ## grouping genes into K subgroups from low to high expression
  group_id = rep(1:4, each = 1000)

  boxplot(sebetahat_ols ~ group_id, xlab = "expression level", ylab = "sebetahat", main = "OLS")
```



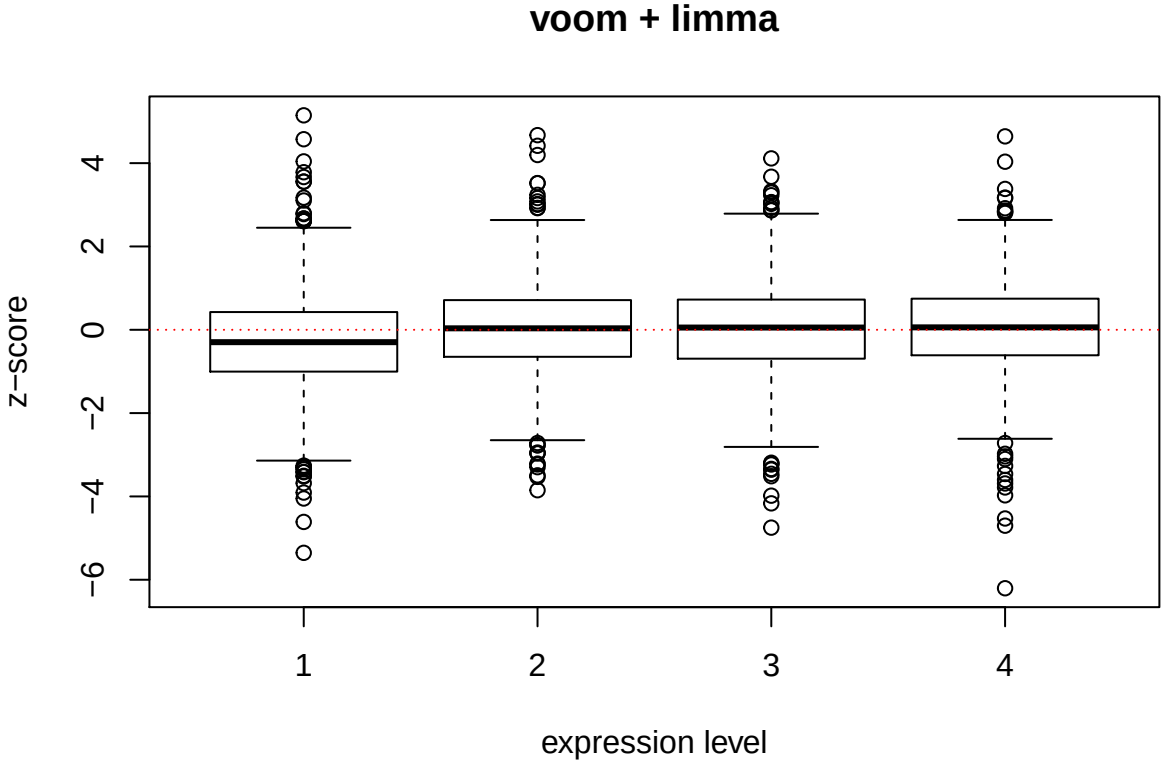

```
boxplot(z_ols ~ group_id, xlab = "expression level", ylab = "z-score", main = "OLS")
abline(0, 0, lty = 3, col = "red")
```



```r
boxplot(sebetahat_voom ~ group_id, xlab = "expression level", ylab = "sebetahat", main = "voom + limma
```




```r
boxplot(z_voom ~ group_id, xlab = "expression level", ylab = "z-score", main = "voom + limma")
abline(0, 0, lty = 3, col = "red")
```



Null data, 5 hearts vs 5 hearts

Now we sample 5 hearts vs 5 hearts, and order the genes from low expression to high expression.

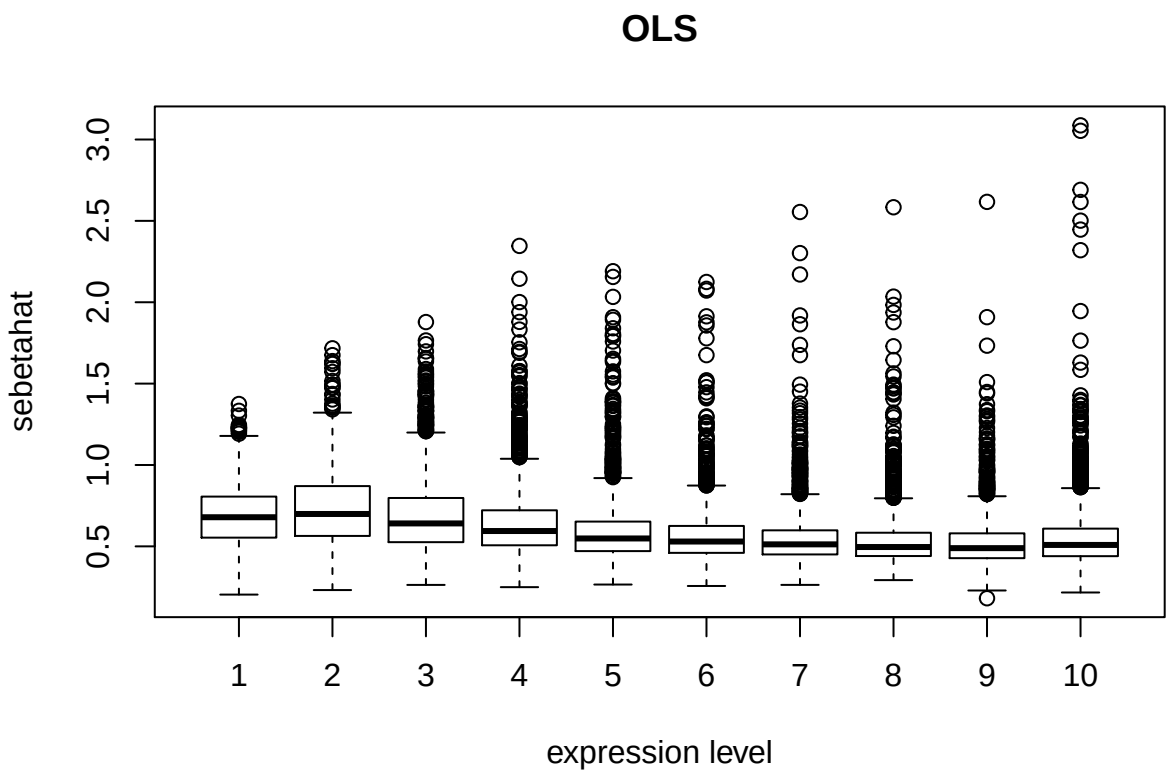
```
source("../code/datamaker_gtex.R")
dat = datamaker_gtex(tissue = "heart", Nsample = 5)
counts = dat$counts
condition = dat$condition
```

With the count matrix and the condition vector, we could get a $\hat{\beta}$ and $\hat{s}$, as well as a z-score $= \hat{\beta}/\hat{s}$ for each gene by both OLS and voom + limma pipeline.

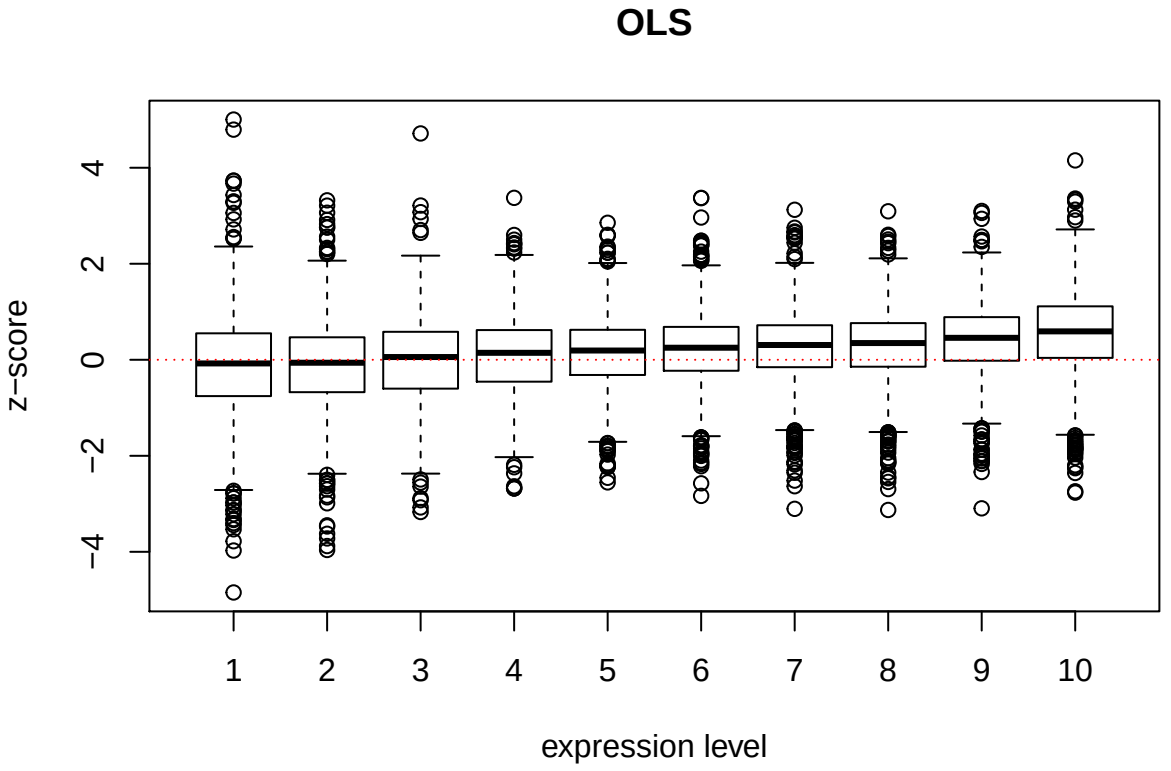
```
source("../code/fit_method.R")

    ## effect size and standard error estimated by OLS
    log_counts = log2(counts + 1)
    ols_fit <- get_ols(log_counts = log_counts, condition = condition)
    betahat_ols = ols_fit$betahat
    sebetahat_ols = ols_fit$sebetahat
    df_ols = ols_fit$df
    z_ols = betahat_ols / sebetahat_ols

    ## effect size and measurement error estimated by voom + limma
    voom_fit = voom_transform(counts, condition)
    betahat_voom = voom_fit$betahat
    sebetahat_voom = voom_fit$sebetahat
    df_voom = voom_fit$df
    z_voom = betahat_voom / sebetahat_voom
```

We group the genes to 10 equally-sized bins, the first bin consists of the 10% lowest expressed genes, and the tenth bin the 10% highest expressed ones. Then we plot $\hat{s}$ and z-score with respect to the expression level.

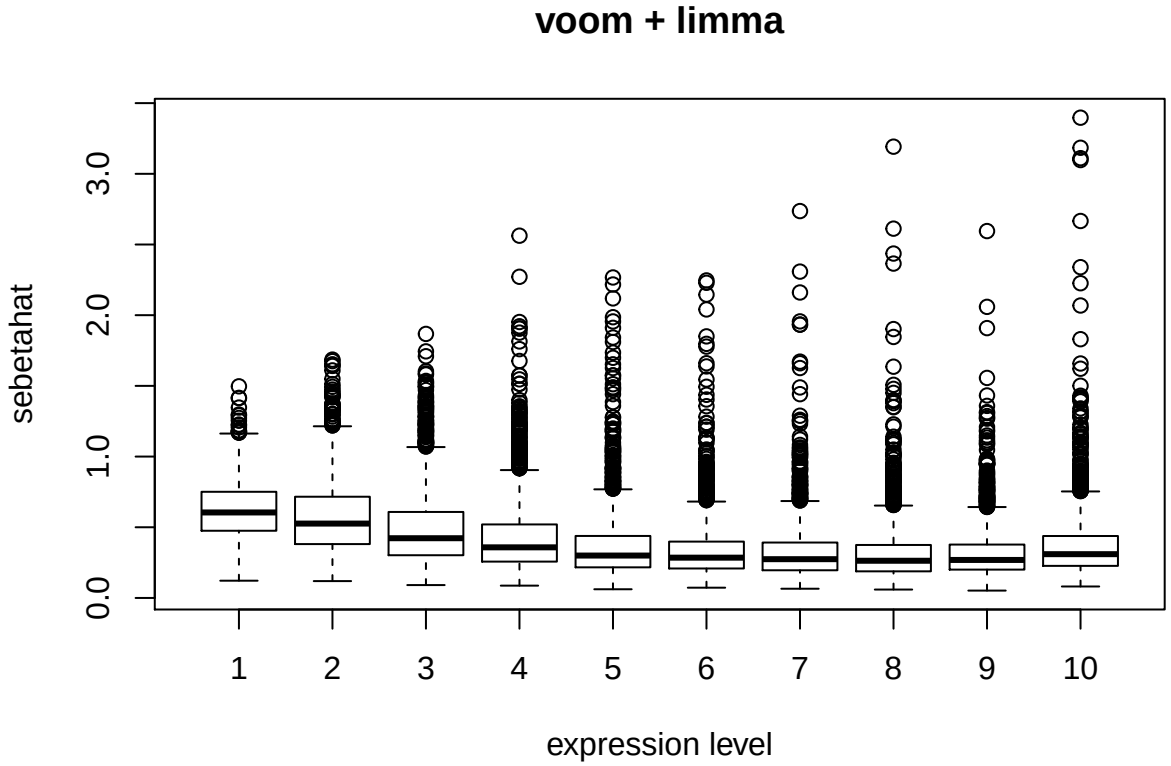
```
    ## grouping genes into K subgroups from low to high expression
    K = 10
    group_size = floor(dim(counts)[1]/K)
    group_id = rep(K, dim(counts)[1])
    group_id[1:(K * group_size)] = rep(1:K, each = group_size)

  boxplot(sebetahat_ols ~ group_id, xlab = "expression level", ylab = "sebetahat", main = "OLS")
```



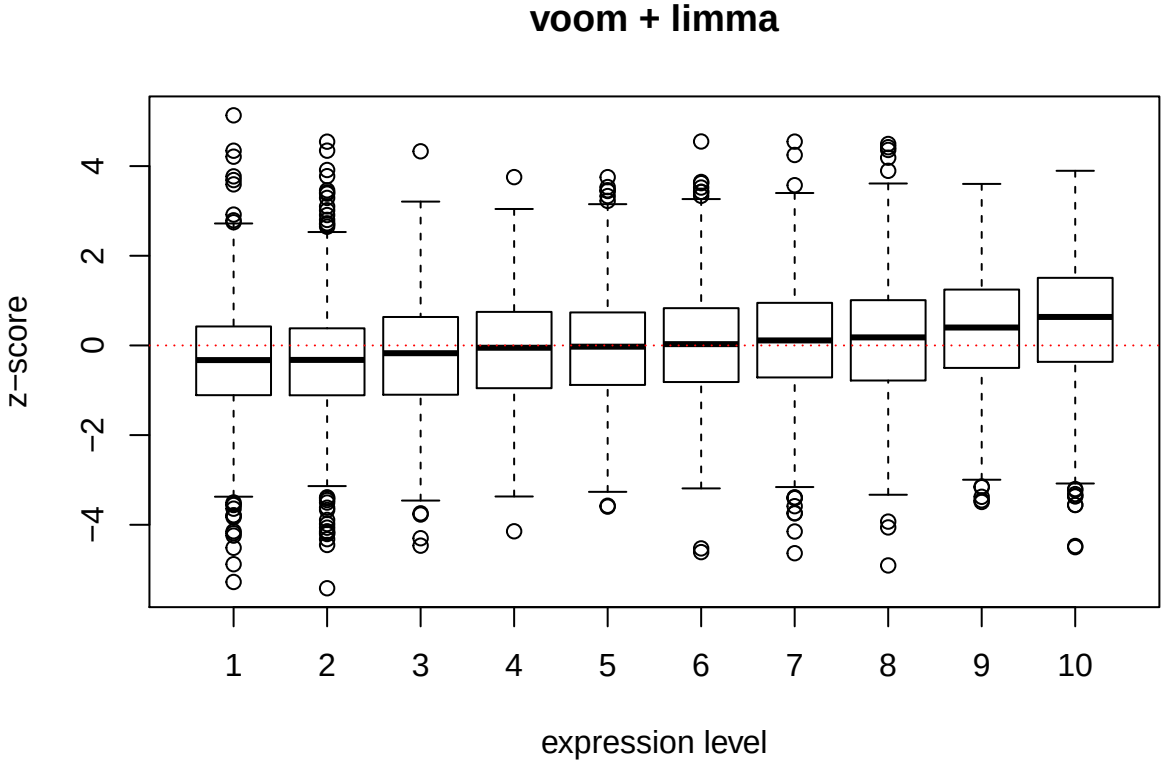

```r
boxplot(z_ols ~ group_id, xlab = "expression level", ylab = "z-score", main = "OLS")
abline(0, 0, lty = 3, col = "red")
```



```r
boxplot(sebetahat_voom ~ group_id, xlab = "expression level", ylab = "sebetahat", main = "voom + limma
```




```r
boxplot(z_voom ~ group_id, xlab = "expression level", ylab = "z-score", main = "voom + limma")
abline(0, 0, lty = 3, col = "red")
```



Alternative data, 5 hearts vs 5 muscles

Then we sample 5 hearts vs 5 muscles, and order the genes from low expression to high expression.

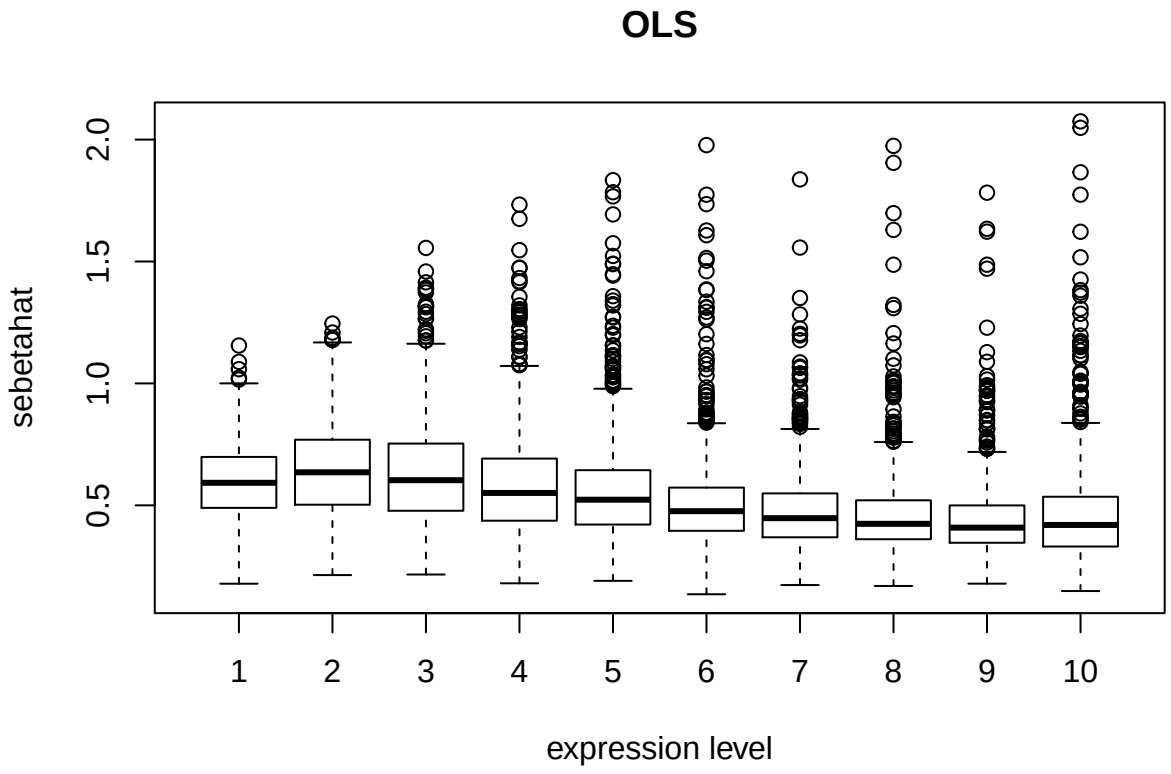
```
source("../code/datamaker_gtex.R")
dat = datamaker_gtex(tissue = c("heart", "muscle"), Nsample = 5)
counts = dat$counts
condition = dat$condition
```

With the count matrix and the condition vector, we could get a $\hat{\beta}$ and $\hat{s}$, as well as a z-score $= \hat{\beta}/\hat{s}$ for each gene by both OLS and voom + limma pipeline.

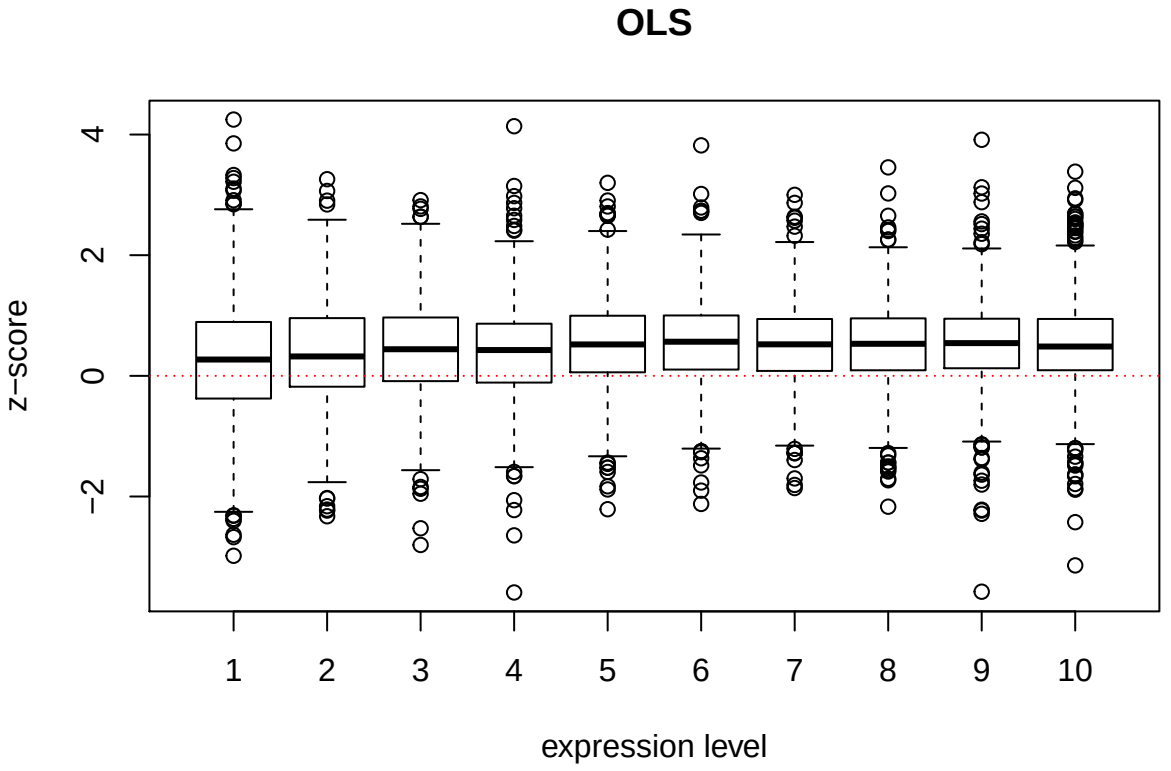
```
source("../code/fit_method.R")

    ## effect size and standard error estimated by OLS
    log_counts = log2(counts + 1)
    ols_fit <- get_ols(log_counts = log_counts, condition = condition)
    betahat_ols = ols_fit$betahat
    sebetahat_ols = ols_fit$sebetahat
    df_ols = ols_fit$df
    z_ols = betahat_ols / sebetahat_ols

    ## effect size and measurement error estimated by voom + limma
    voom_fit = voom_transform(counts = counts, condition = condition)
    betahat_voom = voom_fit$betahat
    sebetahat_voom = voom_fit$sebetahat
    df_voom = voom_fit$df
    z_voom = betahat_voom / sebetahat_voom
```

We group the genes to 10 equally-sized bins, the first bin consists of the 10% lowest expressed genes, and the tenth bin the 10% highest expressed ones. Then we plot $\hat{s}$ and z-score with respect to the expression level.

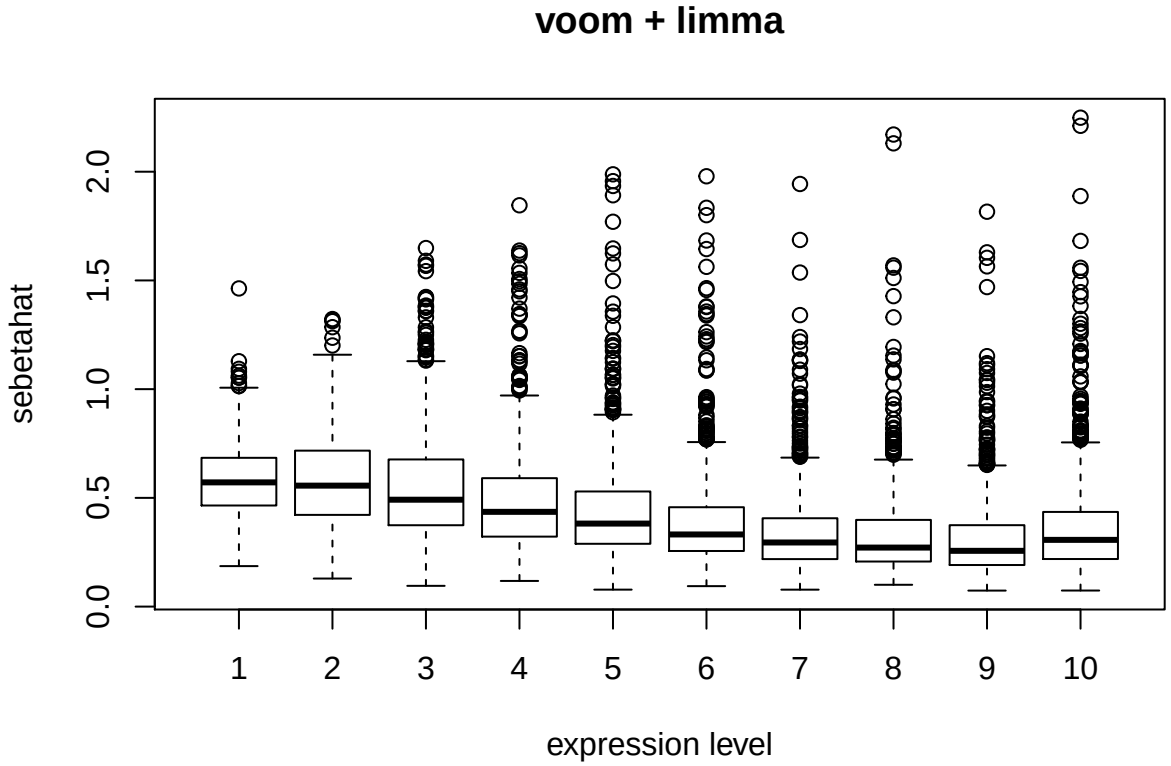
```
    ## grouping genes into K subgroups from low to high expression
    K = 10
    group_size = floor(dim(counts)[1]/K)
    group_id = rep(K, dim(counts)[1])
    group_id[1:(K * group_size)] = rep(1:K, each = group_size)

  boxplot(sebetahat_ols ~ group_id, xlab = "expression level", ylab = "sebetahat", main = "OLS")
```




```r
boxplot(z_ols ~ group_id, xlab = "expression level", ylab = "z-score", main = "OLS")
abline(0, 0, lty = 3, col = "red")
```



```
boxplot(sebetahat_voom ~ group_id, xlab = "expression level", ylab = "sebetahat", main = "voom + limma
```




```
boxplot(z_voom ~ group_id, xlab = "expression level", ylab = "z-score", main = "voom + limma")
abline(0, 0, lty = 3, col = "red")
```