homework3

```
library(IsoformSwitchAnalyzeR)
```

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

```
## Loading required package: DEXSeq
```

```
## Loading required package: BiocParallel
```

```
## Loading required package: Biobase
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## The following object is masked from 'package:limma':
##
##     plotMA
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter,
##     Find, get, grep, grepl, intersect, is.unsorted, lapply, Map,
##     mapply, match, mget, order, paste, pmax, pmax.int, pmin,
##     pmin.int, Position, rank, rbind, Reduce, rownames, sapply,
##     setdiff, sort, table, tapply, union, unique, unsplit, which,
##     which.max, which.min
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
## Loading required package: SummarizedExperiment
```

```
## Loading required package: GenomicRanges
```

```
## Loading required package: stats4
```

```
## Loading required package: S4Vectors
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:base':
##
##     expand.grid
## Loading required package: IRanges
## Loading required package: GenomeInfoDb
## Loading required package: DelayedArray
## Loading required package: matrixStats
##
## Attaching package: 'matrixStats'
## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians
##
## Attaching package: 'DelayedArray'
## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges
## The following objects are masked from 'package:base':
##
##     aperm, apply, rowsum
## Loading required package: DESeq2
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
## Loading required package: AnnotationDbi
## Loading required package: RColorBrewer
## Loading required package: ggplot2
library(tidyverse)
## -- Attaching packages ----------------------------------------------------- tidyverse 1.2.1 --
## v tibble  2.1.1     v purrr   0.3.2
## v tidyr   0.8.3     v dplyr   0.8.1
## v readr   1.3.1     v stringr 1.4.0
## v tibble  2.1.1     v forcats 0.4.0
## -- Conflicts -------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::collapse()   masks IRanges::collapse()
## x dplyr::combine()    masks Biobase::combine(), BiocGenerics::combine()
## x dplyr::count()      masks matrixStats::count()
## x dplyr::desc()       masks IRanges::desc()
## x tidyr::expand()     masks S4Vectors::expand()
## x dplyr::filter()     masks stats::filter()
## x dplyr::first()      masks S4Vectors::first()
## x dplyr::lag()        masks stats::lag()
## x ggplot2::Position() masks BiocGenerics::Position(), base::Position()
```

```
## x purrr::reduce()     masks GenomicRanges::reduce(), IRanges::reduce()
## x dplyr::rename()     masks S4Vectors::rename()
## x dplyr::select()     masks AnnotationDbi::select()
## x purrr::simplify()   masks DelayedArray::simplify()
## x dplyr::slice()      masks IRanges::slice()
```

```
library(pheatmap)
```

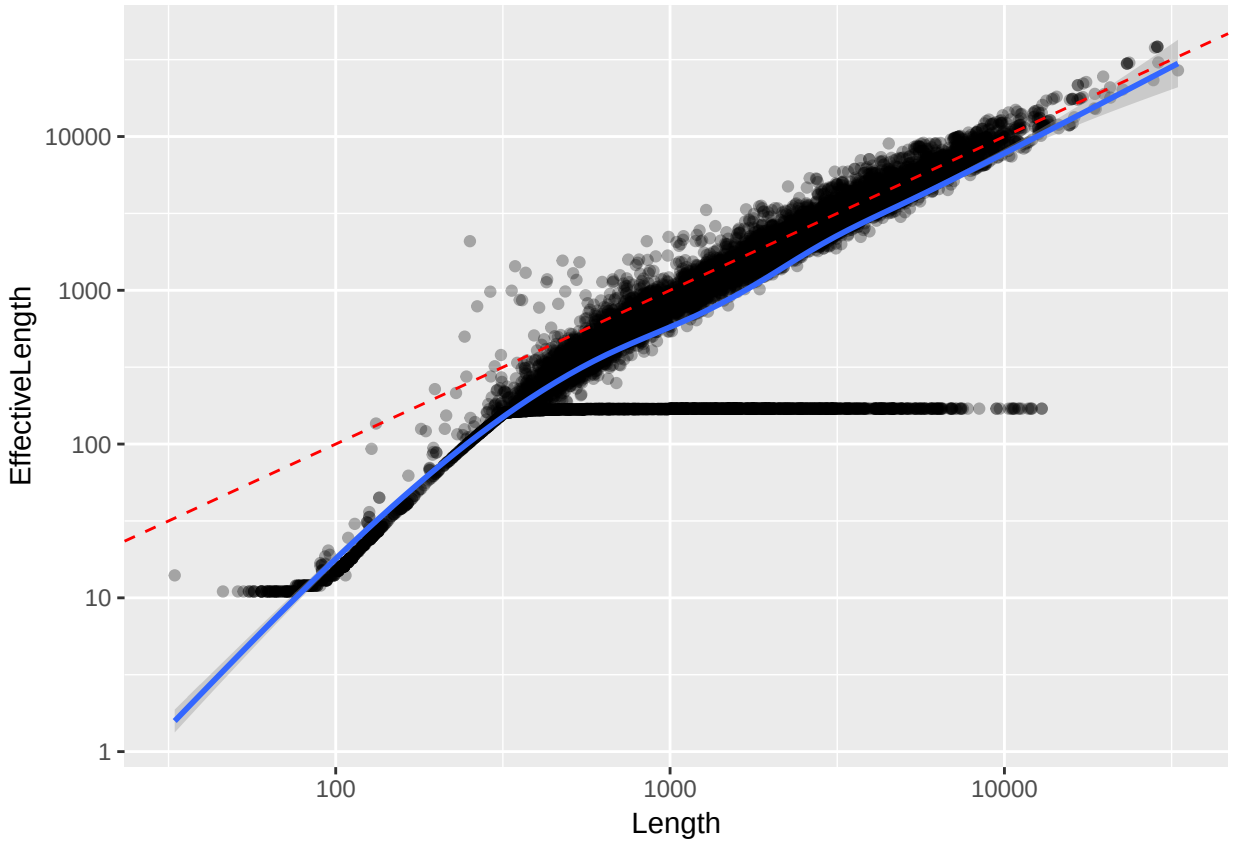
Question 1.1

```
setwd("/home/nuttapong/Desktop/block4/hightp/homework3/HW3_combined_handout/")
wt1_quant <- read_tsv("./HW3_combined_handout/salmon_result_part1/salmon_result_part1/WT1/quant.sf")
```

```
## Parsed with column specification:
## cols(
##   Name = col_character(),
##   Length = col_double(),
##   EffectiveLength = col_double(),
##   TPM = col_double(),
##   NumReads = col_double()
## )
```

```
wt1_quant %>% ggplot(aes(x=Length, y=EffectiveLength)) +
  scale_x_continuous(trans='log10') + scale_y_continuous(trans='log10')  +
  geom_point(alpha=0.3) +
  geom_smooth() + geom_abline(color = "red", linetype=2)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Question 1.2

Question 1.3

```
setwd("/home/nuttapong/Desktop/block4/hightp/homework3/HW3_combined_handout/")
all_salmons <- importIsoformExpression(parentDir = "./HW3_combined_handout/salmon_result_part1/salmon_r
```

```
## Step 1 of 3: Identifying which algorithm was used...
##     The quantification algorithm used was: Salmon
##     Found 6 quantification file(s) of interest
## Step 2 of 3: Reading data...
## reading in files with read_tsv
## 1 2 3 4 5 6
## Step 3 of 3: Normalizing FPKM/TxPM values via edgeR...
## Done
```

```
salmon_matrix <- as.matrix(all_salmons$abundance[,2:ncol(all_salmons$abundance)])
rownames(salmon_matrix) <- all_salmons$abundance[,1]

transformed_salmon <- log2(salmon_matrix+1)

transformed_salmon[1:4,]
```

```
##                      WT1       WT2       WT3    WTTPA1    WTTPA2 WTTPA3
## TCONS_00000001 0.2973299 0.0000000 0.0000000 0.3822156 0.0000000      0
## TCONS_00000002 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000      0
## TCONS_00000003 0.0000000 0.2984888 0.2253968 1.0124265 0.0000000      0
## TCONS_00003946 0.0392366 0.0000000 0.1913649 0.0000000 0.0564598      0
```

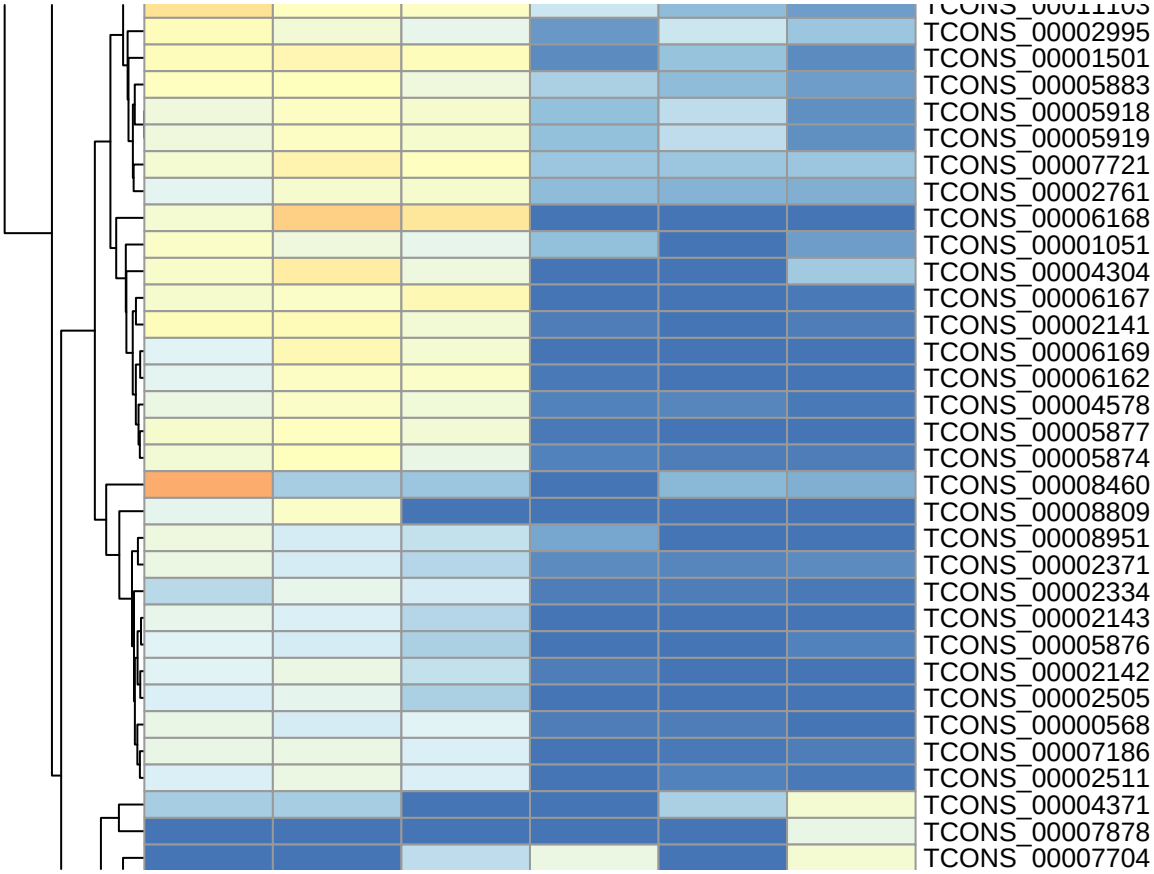
Question 1.4

```
salmon_tibble <- as_tibble(transformed_salmon, rownames=NA)

top100var <- salmon_tibble %>% rownames_to_column() %>% rowwise() %>%
  mutate(variance=var(c(WT1, WT2, WT3, WTTPA1, WTTPA2, WTTPA3))) %>%
  arrange(desc(variance)) %>% slice(1:100)
```

Question 1.5

```
top100var_mat <- as.matrix(top100var[,2:7])
rownames(top100var_mat) <- as.data.frame(top100var)[,1]
pheatmap(top100var_mat, show_rownames = TRUE, cellheight = 10, height = 10)
```


TCONS_00002995
TCONS_00001501
TCONS_00005883
TCONS_00005918
TCONS_00005919
TCONS_00007721
TCONS_00002761
TCONS_00006168
TCONS_00001051
TCONS_00004304
TCONS_00006167
TCONS_00002141
TCONS_00006169
TCONS_00006162
TCONS_00004578
TCONS_00005877
TCONS_00005874
TCONS_00008460
TCONS_00008809
TCONS_00008951
TCONS_00002371
TCONS_00002334
TCONS_00002143
TCONS_00005876
TCONS_00002142
TCONS_00002505
TCONS_00000568
TCONS_00007186
TCONS_00002511
TCONS_00004371
TCONS_00007878
TCONS_00007704