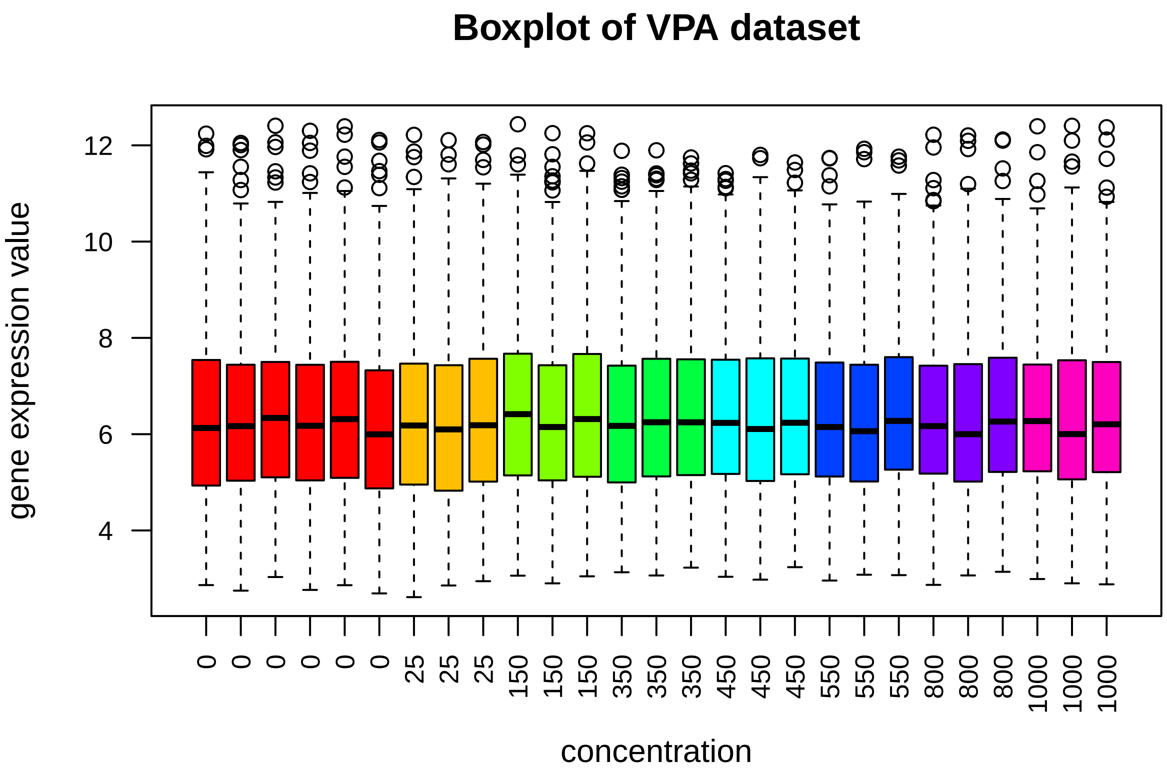
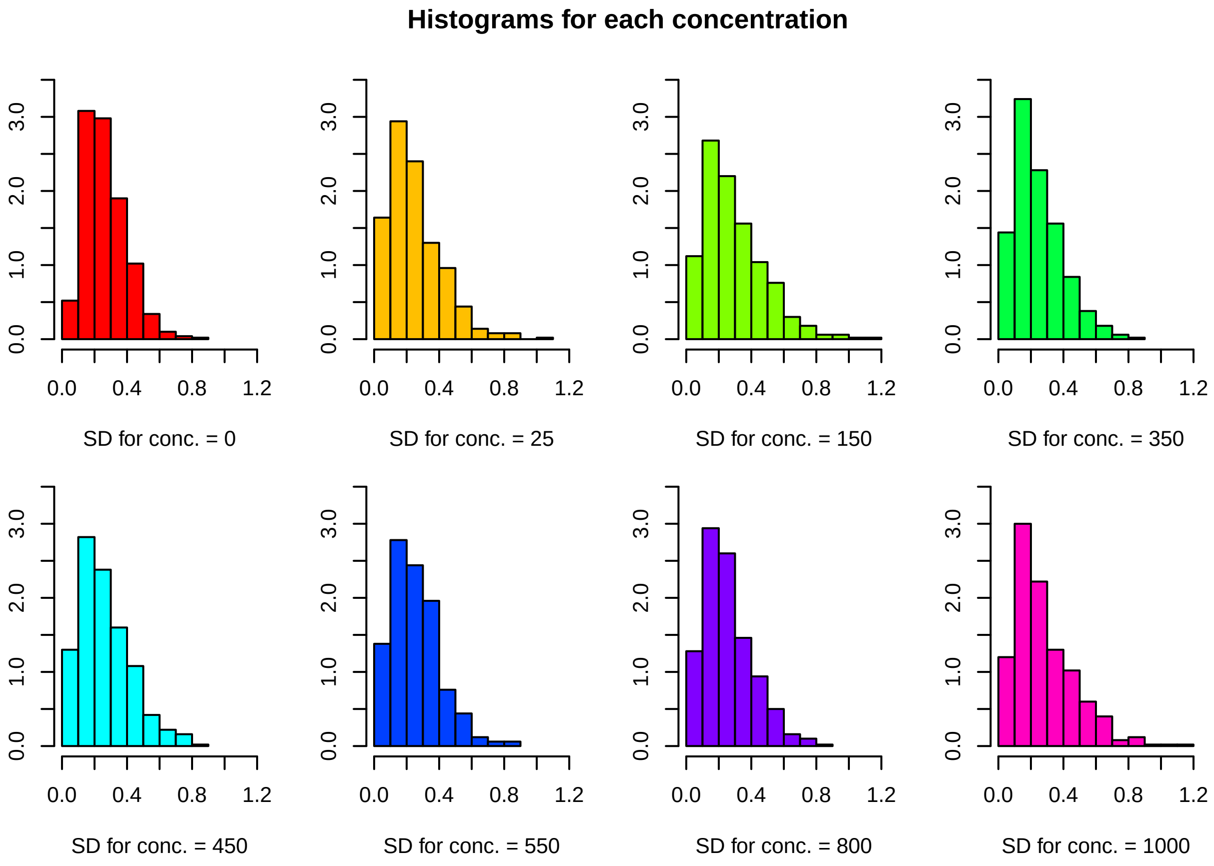
Statistics in Toxicology I - Exercise Sheet 1

```
## Exercise 1

load("VPAData-Random.Rda")

## a)

concentrations <- c(0, 25, 150, 350, 450, 550, 800, 1000)
replicates <- c(rep(concentrations, c(6, rep(3, 7))))

boxplot(randomVPA, main = "Boxplot of VPA dataset", xlab = "concentration",
        ylab = "gene expression value", las = 2, cex.axis = 0.8,
        names = replicates,
        col = rep(rainbow(8), c(6, rep(3, 7))))
```



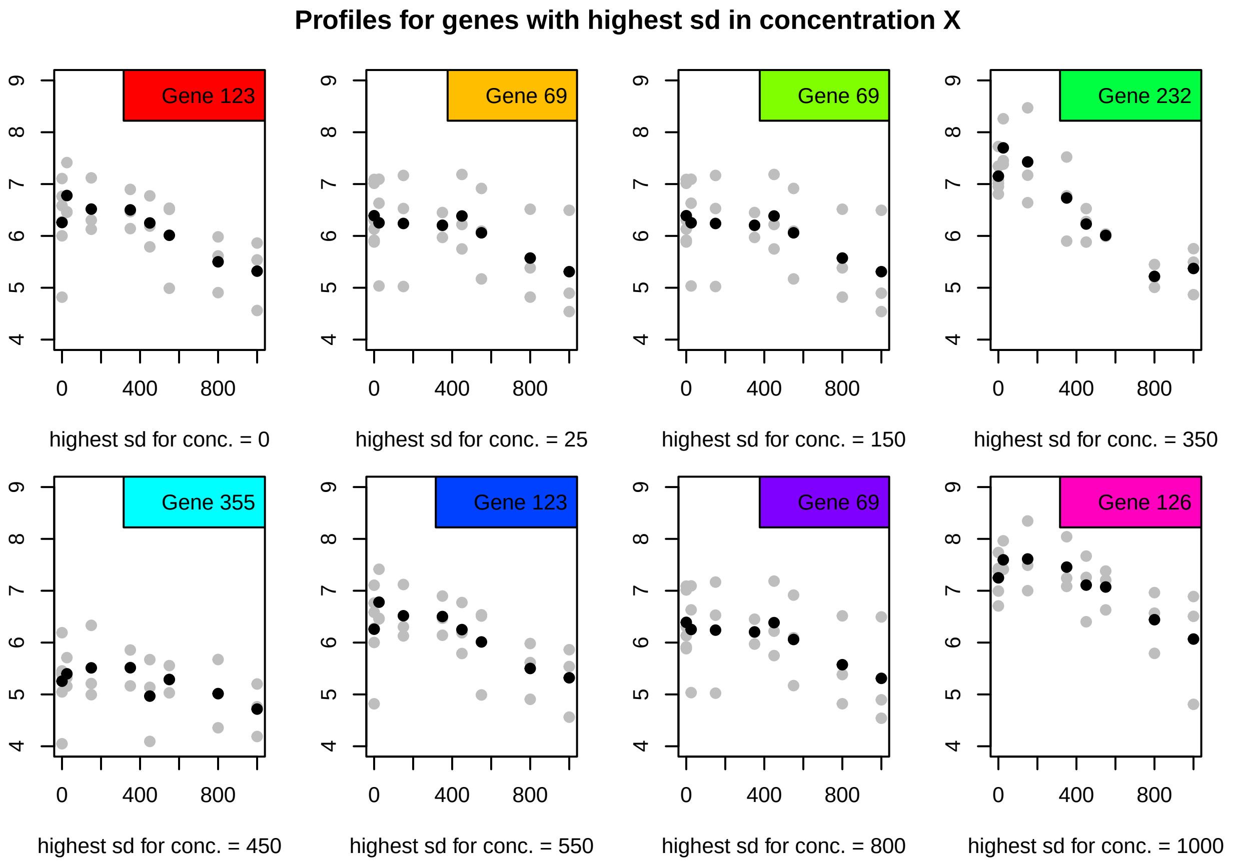
```
## Looks very similar to the example from the lecture, but with fewer data
## points of course.

## b)
```

```r
## sd.concentrations calculates the 8 standard deviations for one gene

sd.concentration <- function(gene) {
  c(sd(gene[1:6]), sd(gene[7:9]), sd(gene[10:12]),
    sd(gene[13:15]), sd(gene[16:18]), sd(gene[19:21]),
    sd(gene[22:24]), sd(gene[25:27]))
}

## Apply to all 500 genes:

deviations <- t(sapply(1:500, FUN = function(x) sd.concentration(randomVPA[x,])))
colnames(deviations) <- c(0, 25, 150, 350, 450, 550, 800, 1000)

## Histogram:
par(mfrow = c(2, 4), mar = c(4, 2, 1, 2), oma = c(0, 0, 2, 0))

apply(matrix(1:8), 1, FUN = function(x) {
  hist(deviations[, x], ylim = c(0, 3.5), xlim = c(0, 1.2),
       freq = FALSE, main = NULL,
       xlab = paste("SD for conc. =", concentrations[x]),
       col = rainbow(8)[x])
  })
title("Histograms for each concentration", outer = TRUE)
```



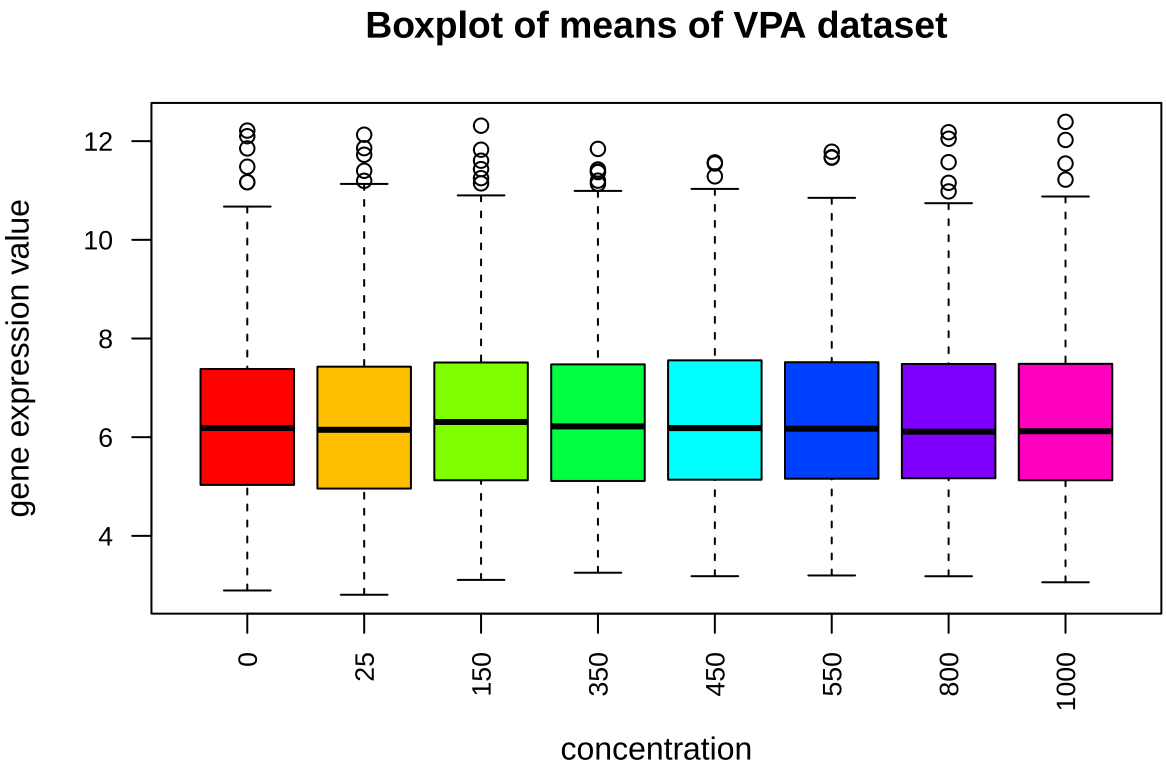
Histograms for each concentration

```r
## c)
## Highest standard deviation:

which.max(deviations[, 1])
which.max(deviations[, 2])


## Plot profiles:

par(mfrow = c(2, 4), mar = c(4, 2, 1, 2), oma = c(0, 0, 2, 0))

apply(matrix(1:8), 1, FUN = function(x) {
  max <- which.max(deviations[, x])
  means <- c(mean(randomVPA[max,1:6]), mean(randomVPA[max,7:9]),
             mean(randomVPA[max,10:12]), mean(randomVPA[max,13:15]),
             mean(randomVPA[max,16:18]), mean(randomVPA[max,19:21]),
             mean(randomVPA[max,22:24]), mean(randomVPA[max,25:27]))

  plot(replicates, randomVPA[max, ],
       pch = 19, col = "grey", ylim = c(4, 9),
       xlab = paste("highest sd for conc. =", concentrations[x]))
  points(concentrations, means, pch = 19)
  legend("topright", legend = paste("Gene", max),
         bg = rainbow(8)[x])
})

title("Profiles for genes with highest sd in concentration X", outer = TRUE)
```



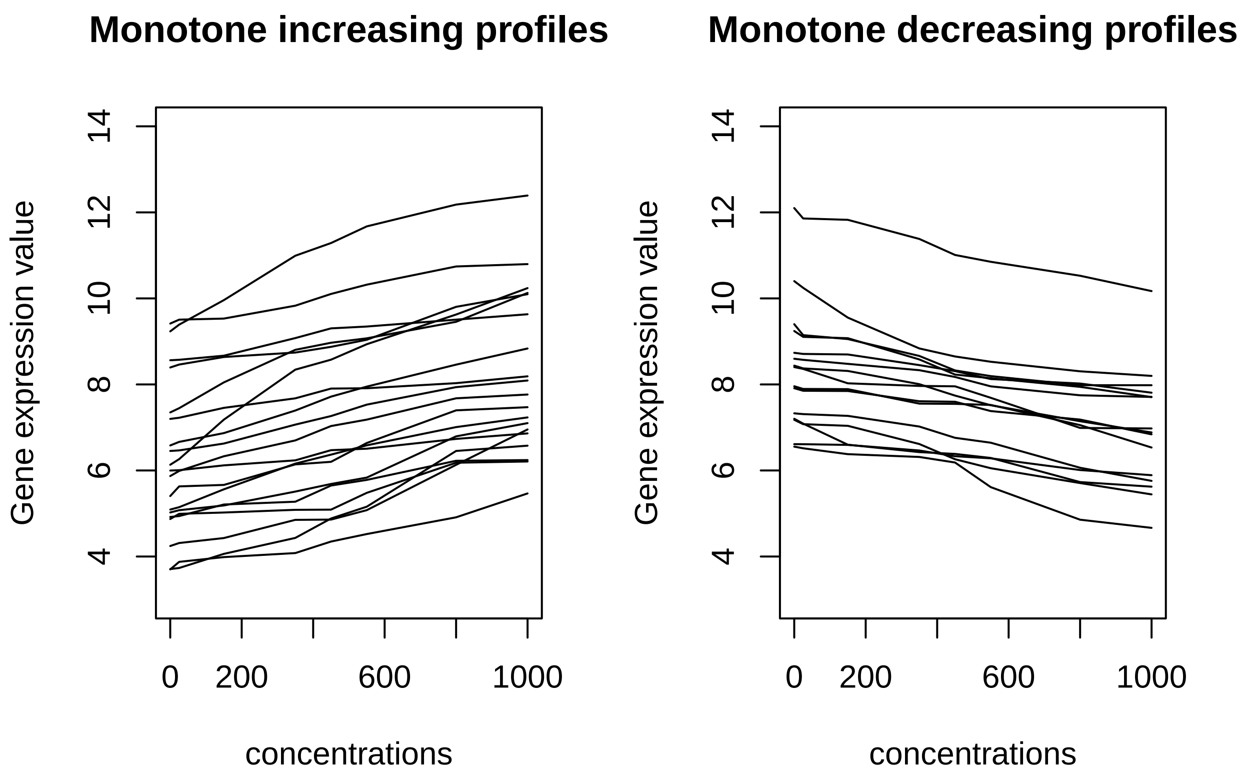
Profiles for genes with highest sd in concentration X

```r
## d)

## calculate means:

mean.concentration <- function(gene) {
  c(mean(gene[1:6]), mean(gene[7:9]), mean(gene[10:12]),
    mean(gene[13:15]), mean(gene[16:18]), mean(gene[19:21]),
    mean(gene[22:24]), mean(gene[25:27]))
}

means <- t(sapply(1:500, FUN = function(x) mean.concentration(randomVPA[x,])))

boxplot(means, main = "Boxplot of means of VPA dataset", xlab = "concentration",
        ylab = "gene expression value", las = 2, cex.axis = 0.8,
        names = concentrations, col = rainbow(8))
```

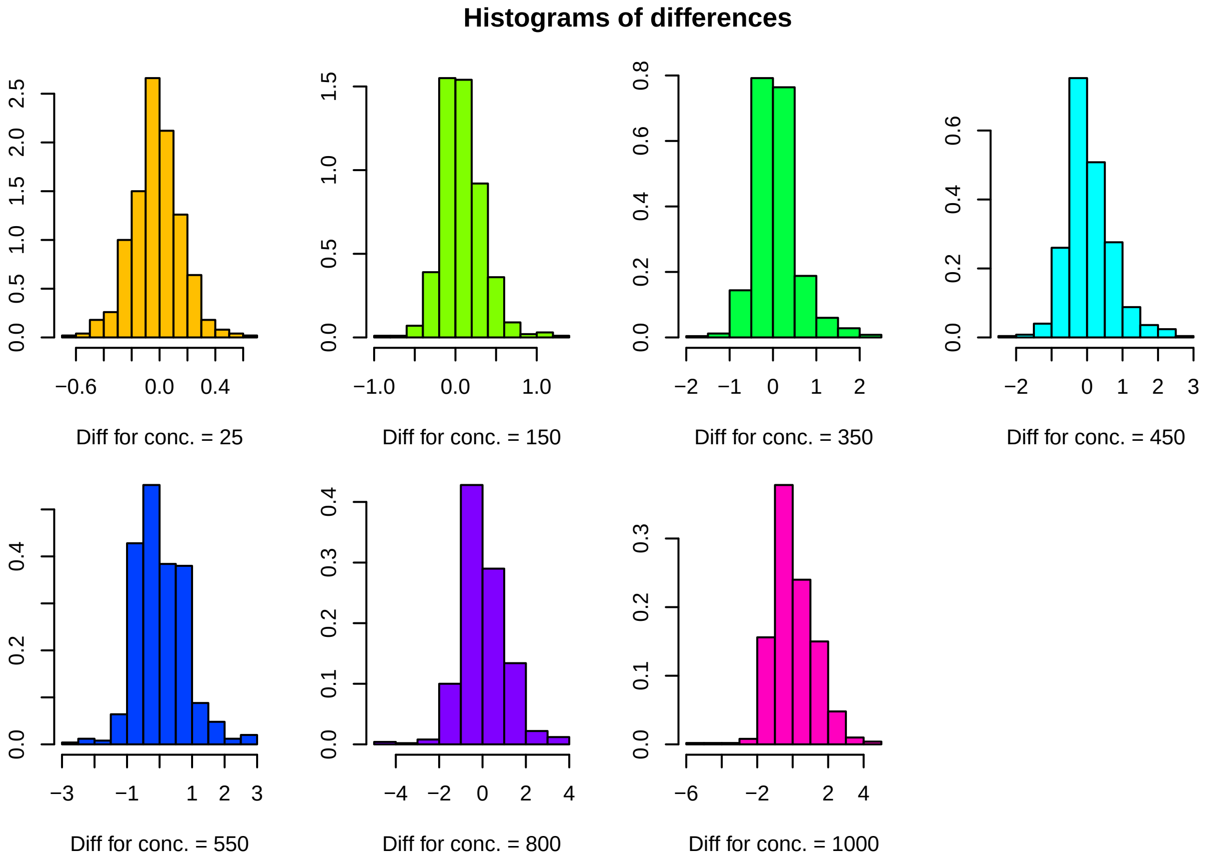


```r
## e)

## Check for Monotonicity using the cummax() for increasing and cummin() for
## decreasing sequences. If the sequence is monotone, it should be identical
## to cummax or cummin:

## Monotone increasing:
inc <- which(sapply(1:500, FUN = function(x) all(means[x,] == cummax(means[x,]))))
## 19 Genes fulfill this

par(mfrow = c(1, 2))

## Profile Plots:
plot(concentrations, means[inc[1],], type = "l", ylim = c(3, 14),
     main = "Monotone increasing profiles", ylab = "Gene expression value")
apply(matrix(2:19), 1,
      FUN = function(x) points(concentrations, means[inc[x],], type = "l"))


## Monotone decreasing:
dec <- which(sapply(1:500, FUN = function(x) all(means[x,] == cummin(means[x,]))))
## 15 Genes fulfill this

## Profile Plots:
plot(concentrations, means[dec[1],], type = "l", ylim = c(3, 14),
     main = "Monotone decreasing profiles", ylab = "Gene expression value")
```

```
apply(matrix(2:15), 1,
      FUN = function(x) points(concentrations, means[dec[x],], type = "l"))
```



```
## f)

## Differences between means of controls and means of positive concentrations:
## Positive Value implies higher expression value than control
## Negative value implies lower expression value than control

Diff <- data.frame(X = means[, 2] - means[, 1])

for(i in 3:8) {
  Diff <- cbind(Diff, means[, i] - means[, 1])
}

## Name columns of dataframe with concentration values
colnames(Diff) <- concentrations[-1]

## Plot Histograms:

par(mfrow = c(2, 4), mar = c(4, 2, 1, 2), oma = c(0, 0, 2, 0))

apply(matrix(1:7), 1, FUN = function(x) {
  hist(Diff[, x],
       freq = FALSE, main = NULL,
       xlab = paste("Diff for conc. =", concentrations[x + 1]),
       col = rainbow(8)[x + 1])
})

title("Histograms of differences", outer = TRUE)
```


Histograms of differences
Diff for conc. = 25
Diff for conc. = 150
Diff for conc. = 350
Diff for conc. = 450
Diff for conc. = 550
Diff for conc. = 800
Diff for conc. = 1000