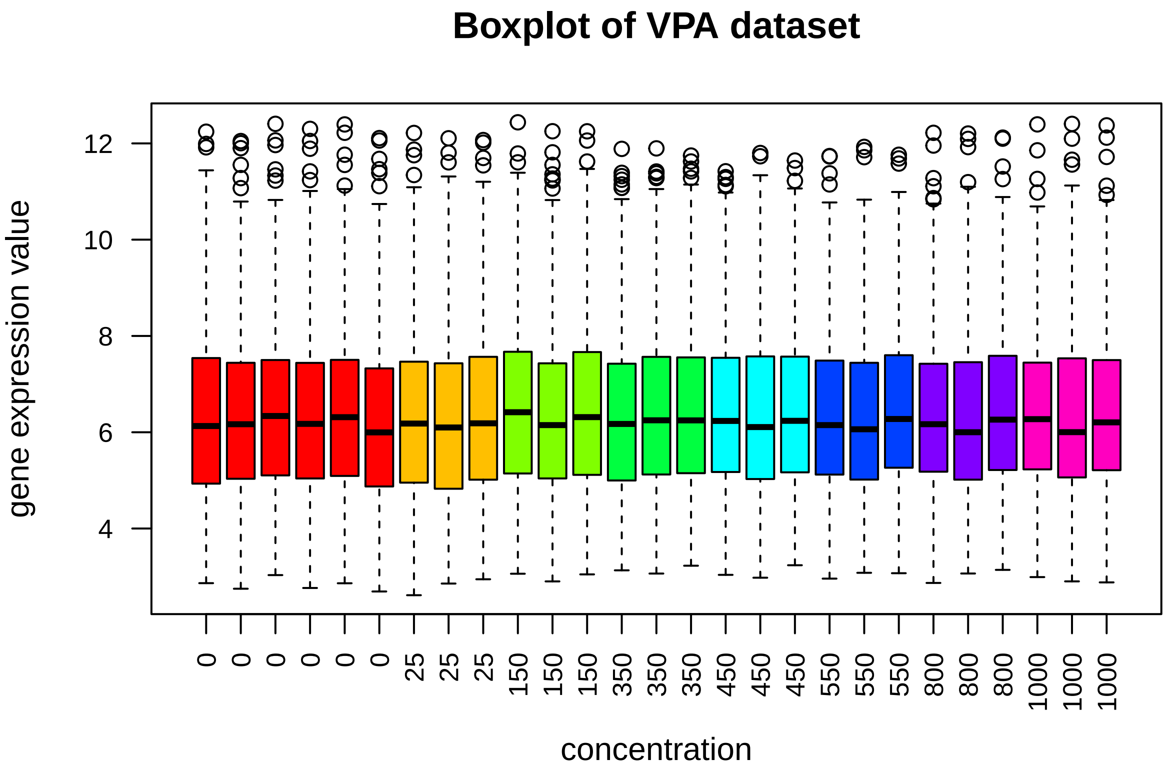
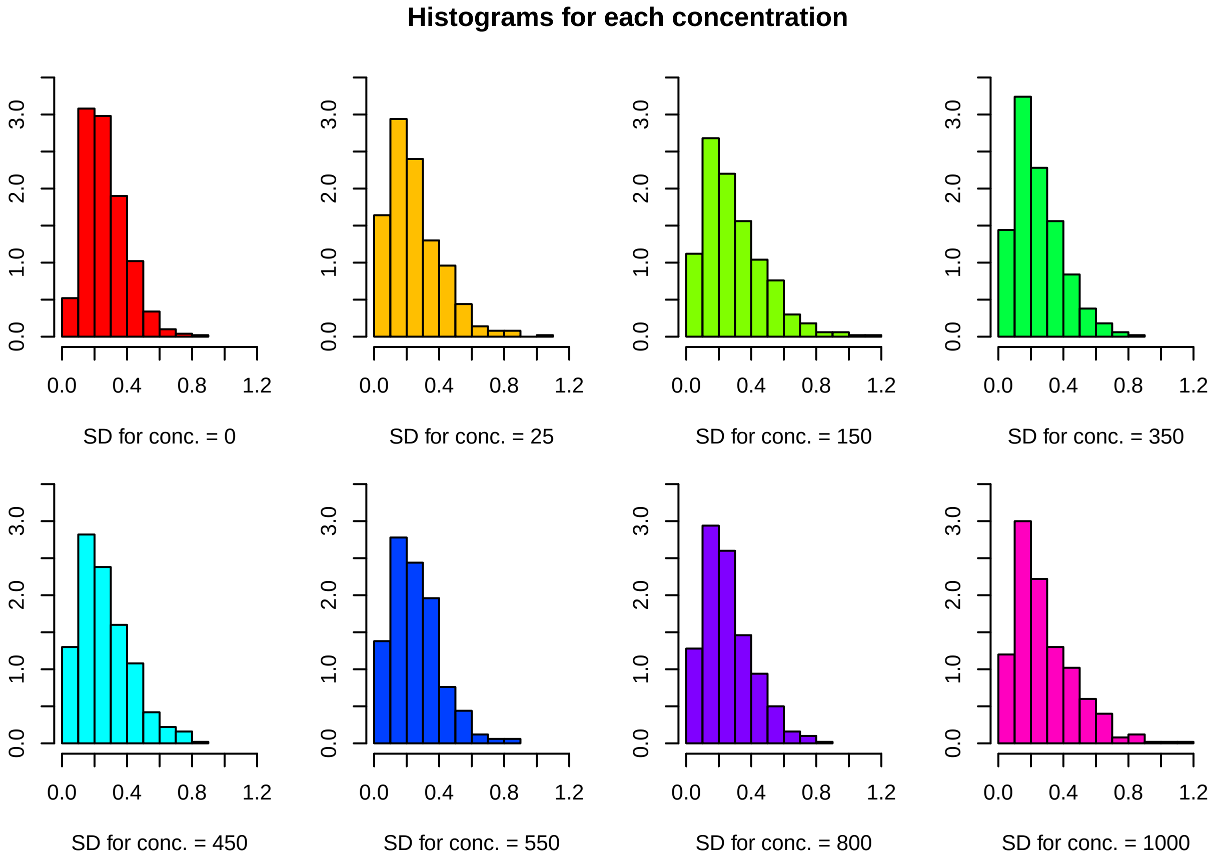
Statistics in Toxicology I - Exercise Sheet 1

Exercise 1: Descriptive analysis of the VPA dataset

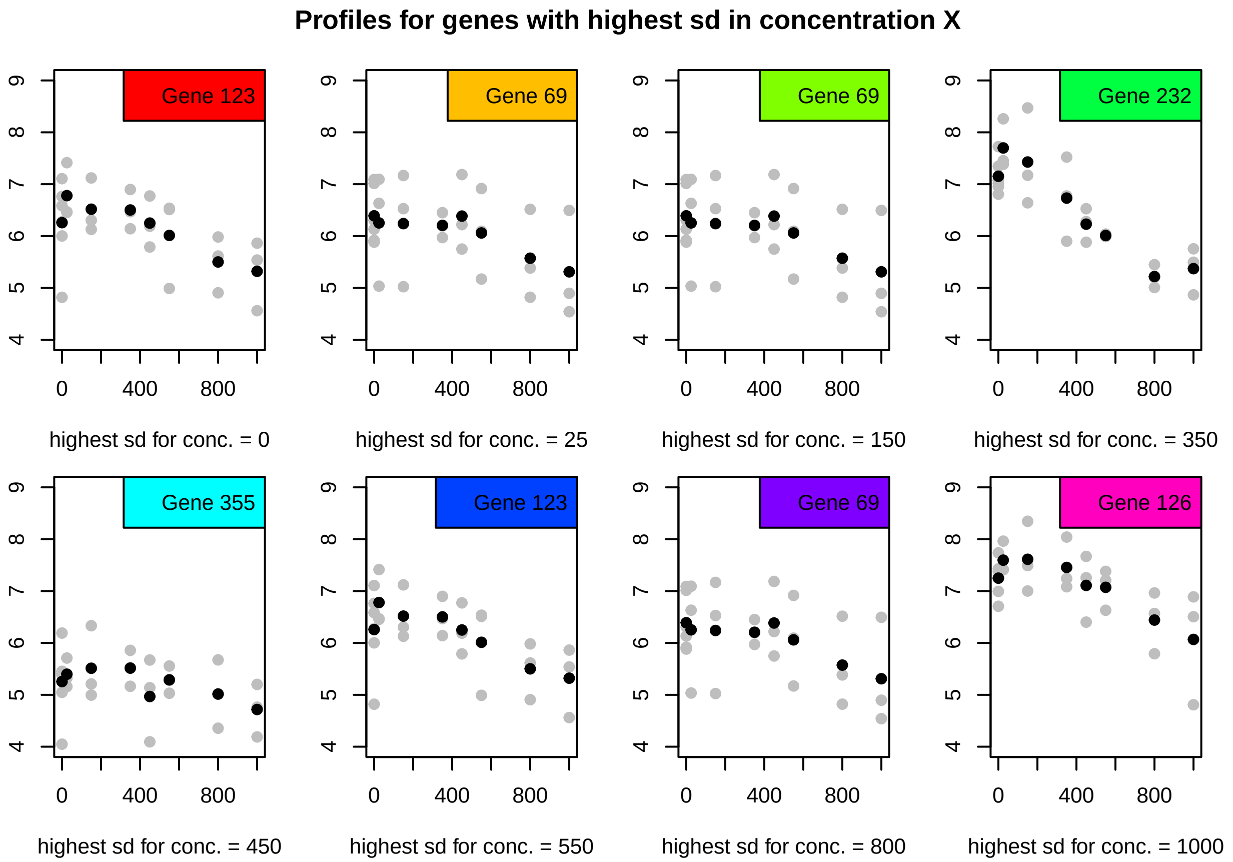
```r
## Exercise 1

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))))
```



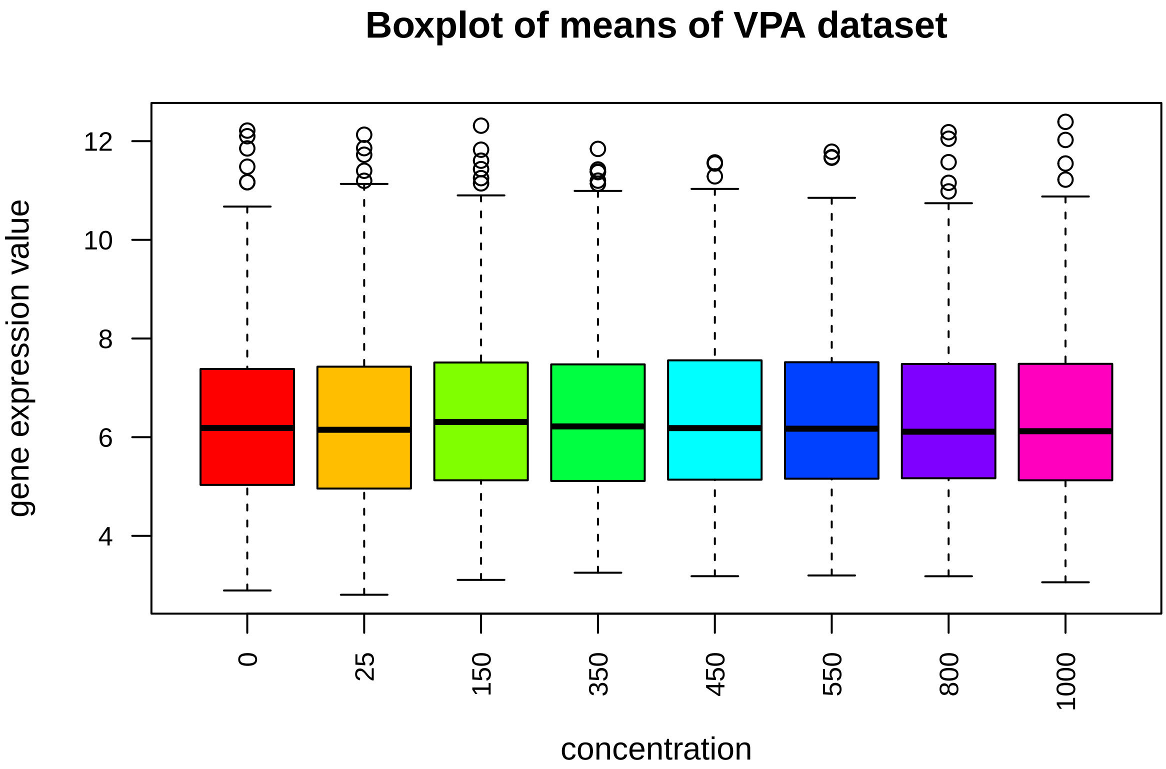
```r
## Looks very similar to the example from the lecture, but with fewer data
## points of course.
## b)

## 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)
```



```
## 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)
```

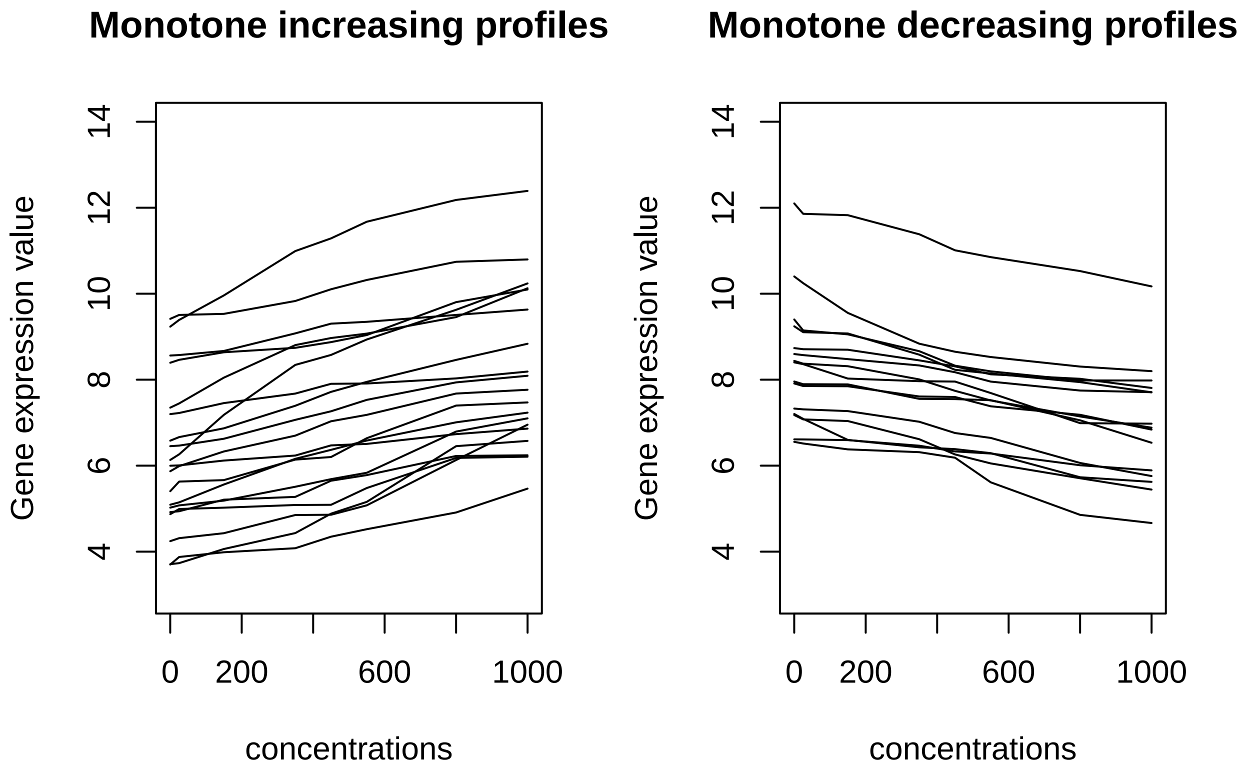


```r
## d)

## 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))
```

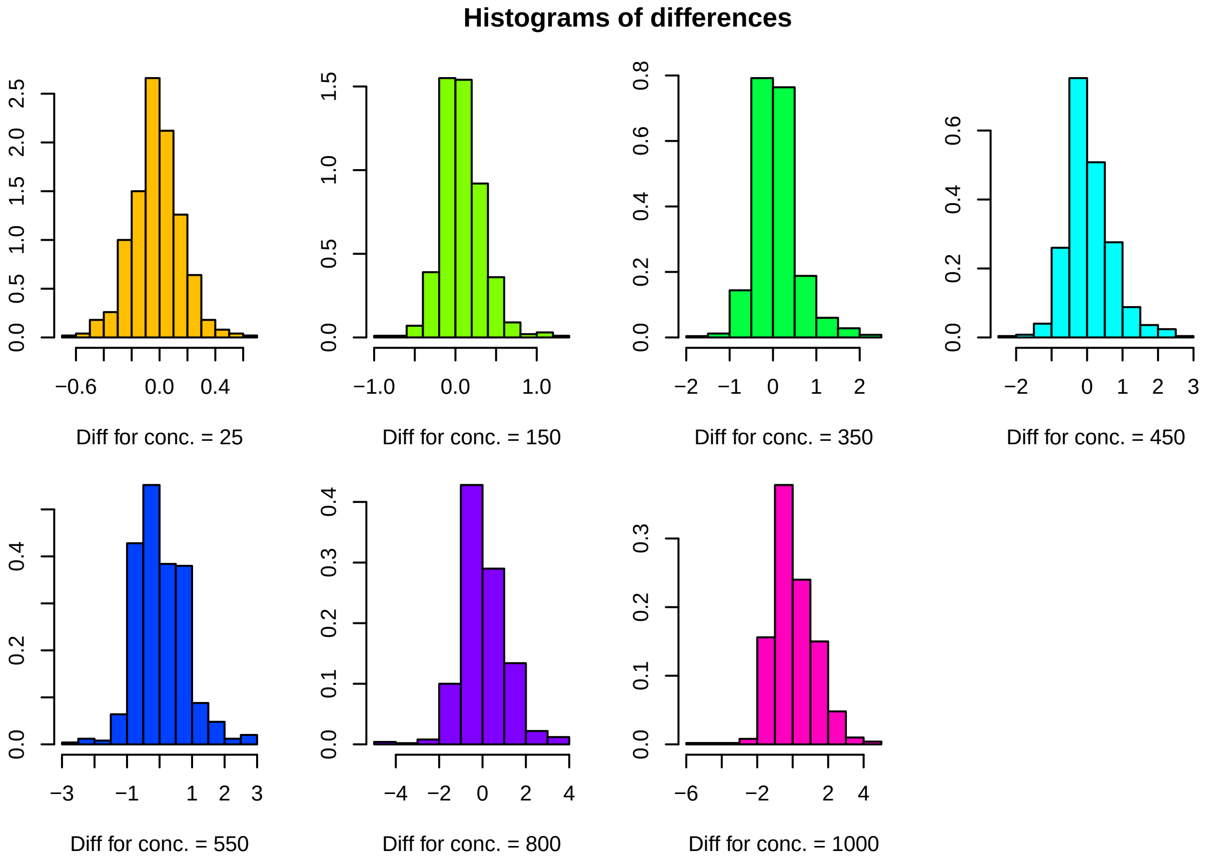


```
## e)

## 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")
```

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



```r
## 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)
```

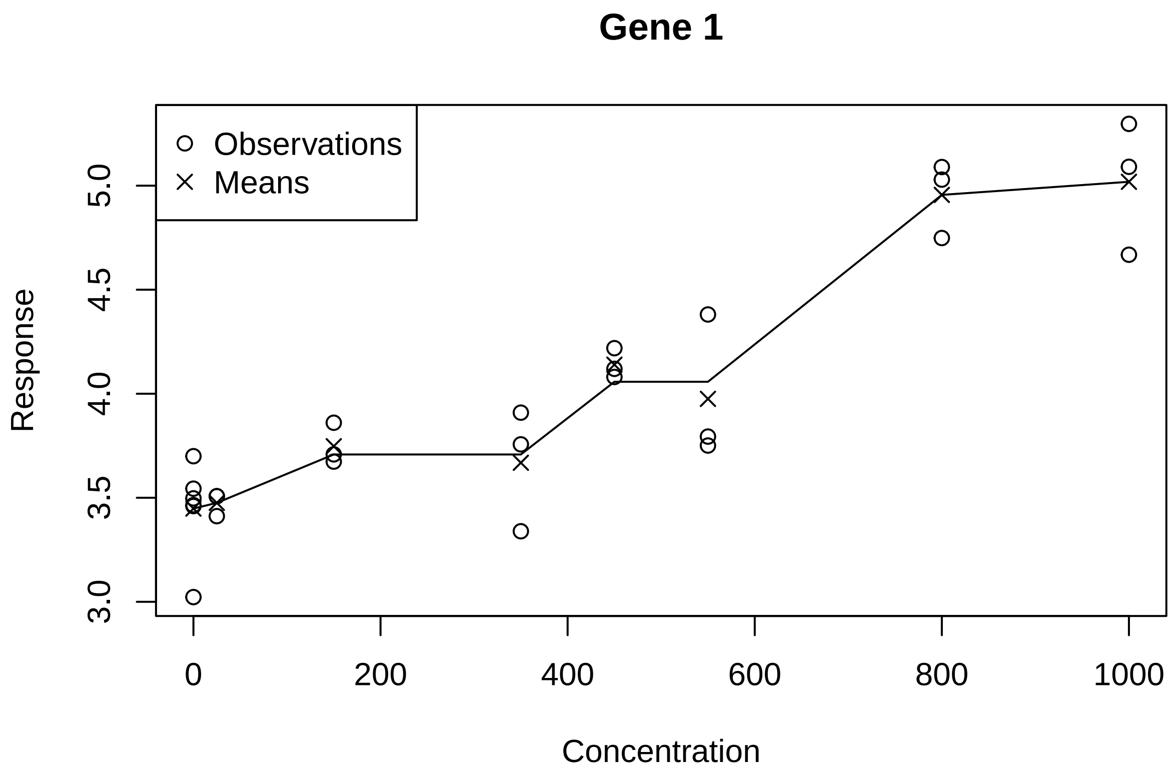


Exercise 3: PAVA II

```r
## (i)

## First, calculate the means for each concentration and add a weights vector:
## Use the mean.concentration function fromm Exercise 1:
meansEx3 <- t(sapply(1:3, FUN = function(x) mean.concentration(VPA.Isotonic[x,])))

weights <- c(6, rep(3, 7))


## Perform isotonic regression with w = weights for first gene:

pava(meansEx3[1,], weights)
```

```
## [1] 3.448079 3.475616 3.708013 3.708013 4.057537 4.057537 4.955646 5.018520
```

```r
## Plot against concentrations and add data points and means:

plot(replicates, VPA.Isotonic[1,],
     ylab = "Response", xlab = "Concentration", main = "Gene 1")
points(concentrations, meansEx3[1, ], pch = 4)
points(concentrations, pava(meansEx3[1,], weights), type = "l")
legend("topleft", legend = c("Observations", "Means"), pch = c(1, 4))
```



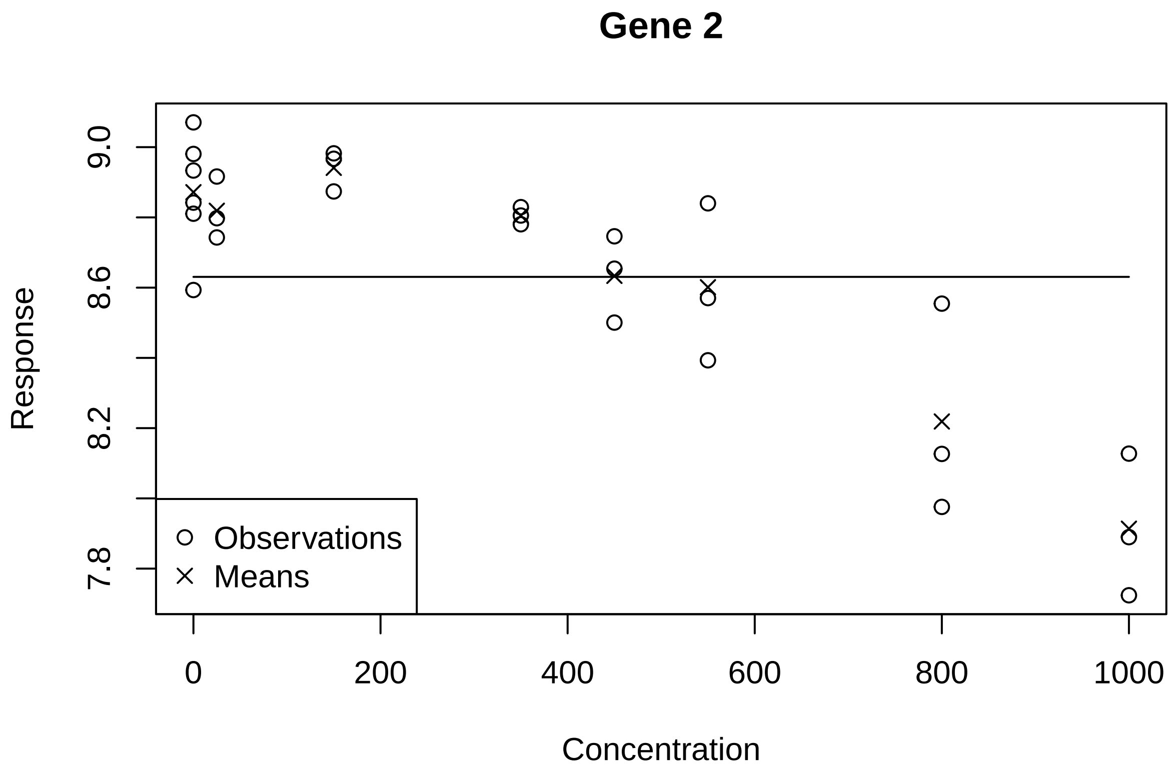
```r
## Isotonic regression works neatly in this example. We see small violations
## between 3rd and 4th concentration as well as between 5th and 6th. All in all
## the assumption of isotonicity seems appropiate.


## Second and third Gene:

## Isotonic regression:

pava(meansEx3[2,], weights)
```

```
## [1] 8.630644 8.630644 8.630644 8.630644 8.630644 8.630644 8.630644 8.630644
```

```
pava(meansEx3[3,], weights)
```

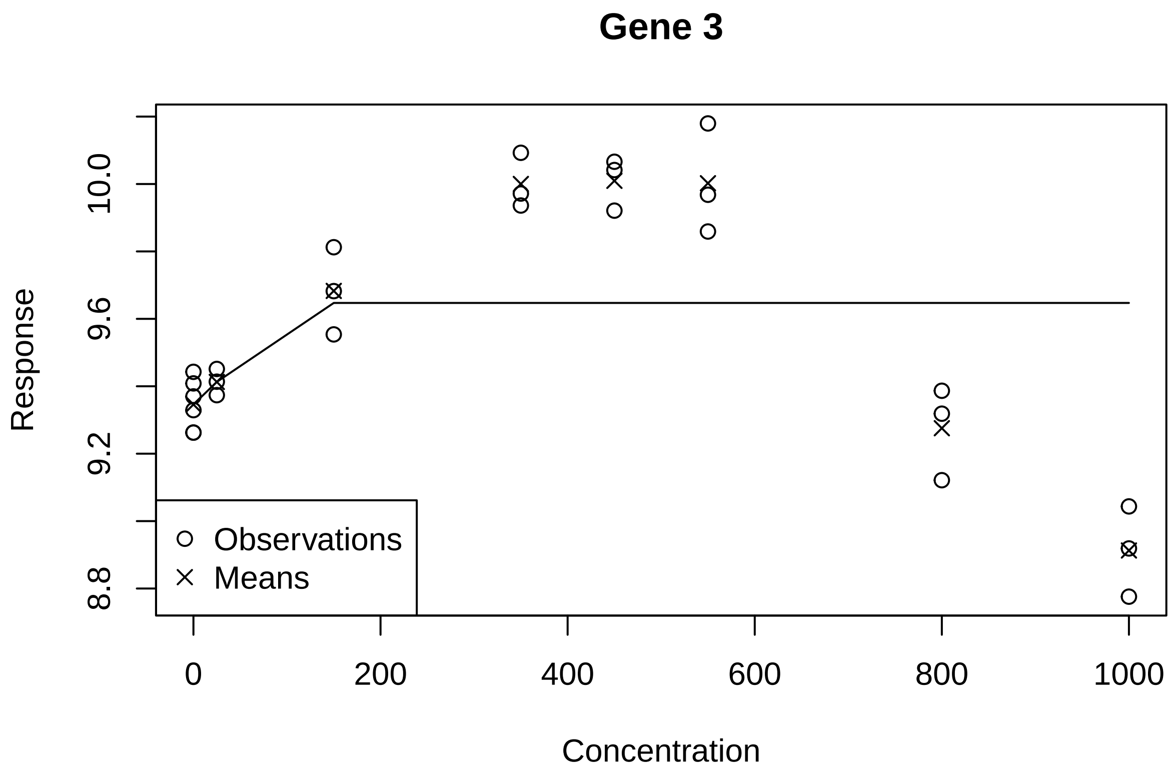
```
## [1] 9.345925 9.412850 9.647181 9.647181 9.647181 9.647181 9.647181 9.647181
```

```
## Plots:

plot(replicates, VPA.Isotonic[2,],
     ylab = "Response", xlab = "Concentration", main = "Gene 2")
points(concentrations, meansEx3[2, ], pch = 4)
points(concentrations, pava(meansEx3[2,], weights), type = "l")
legend("bottomleft", legend = c("Observations", "Means"), pch = c(1, 4))
```

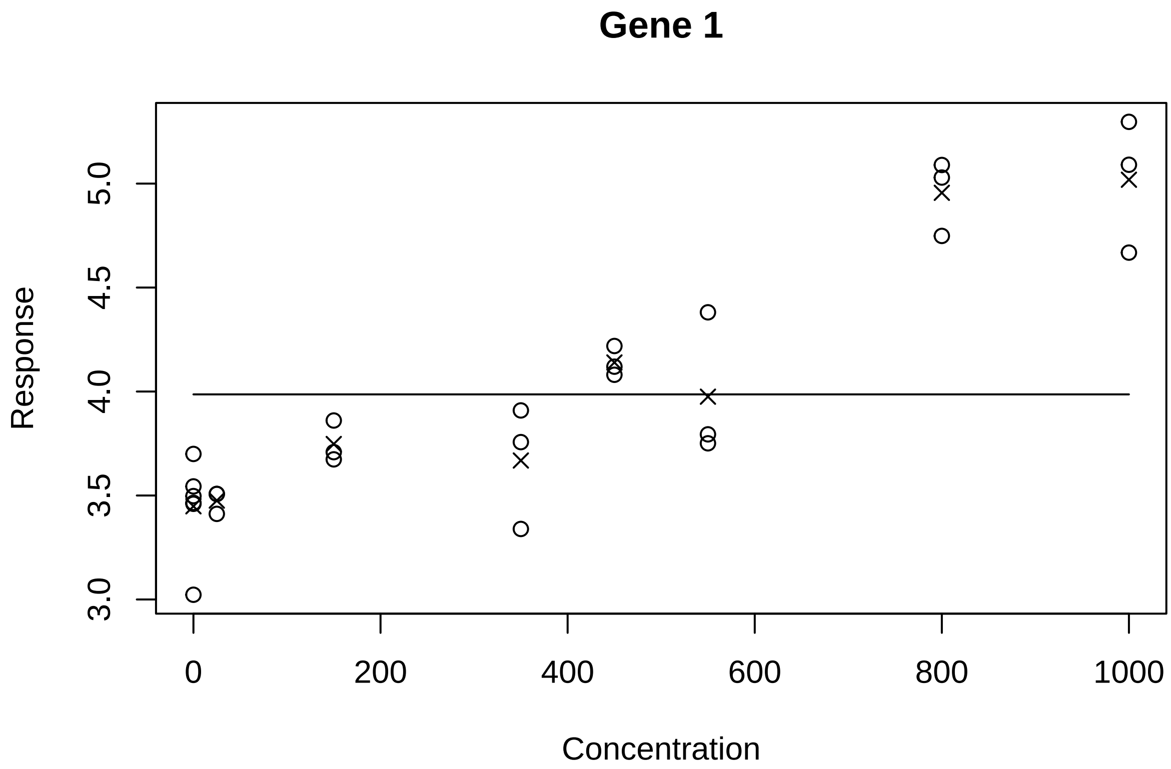


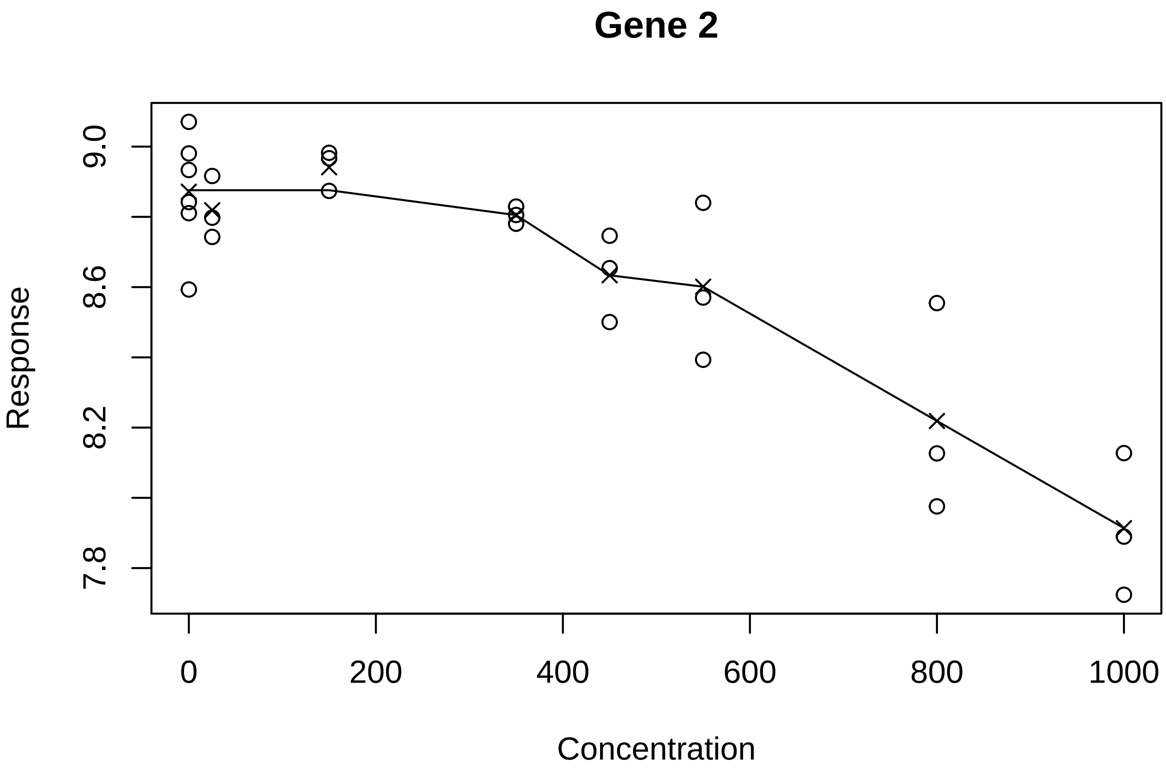
```
plot(replicates, VPA.Isotonic[3,],
     ylab = "Response", xlab = "Concentration", main = "Gene 3")
points(concentrations, meansEx3[3, ], pch = 4)
points(concentrations, pava(meansEx3[3,], weights), type = "l")
```

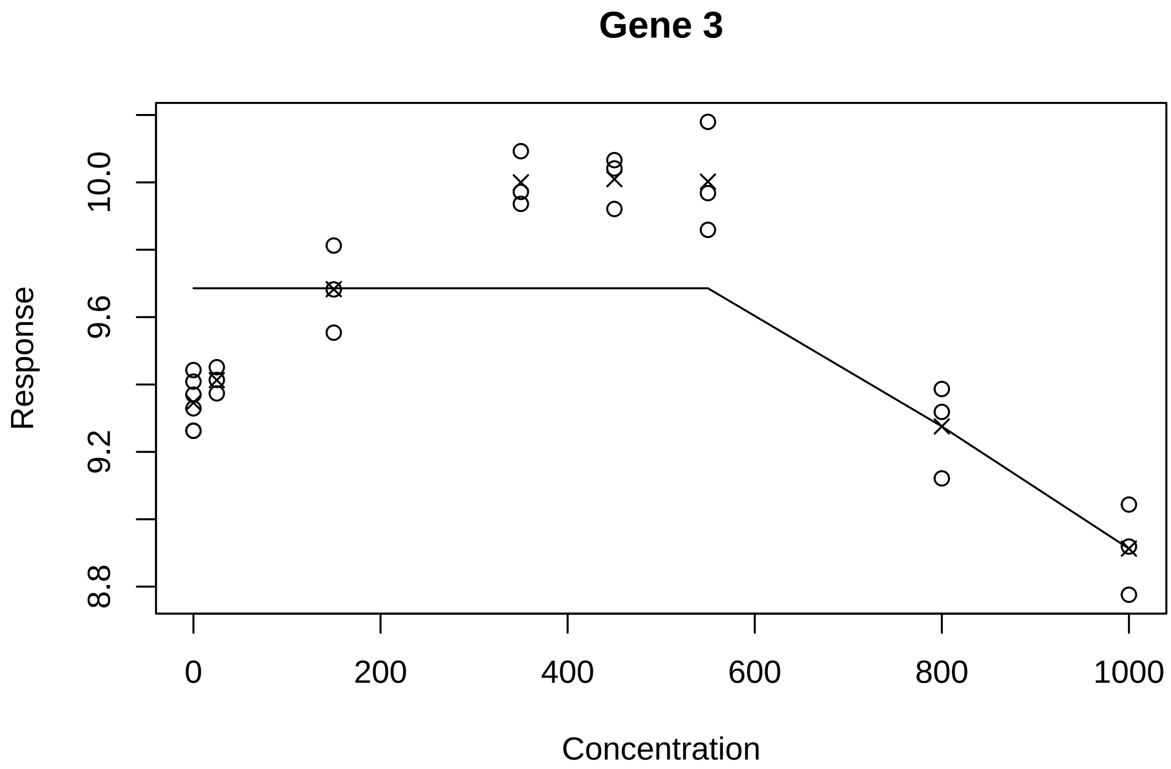
```
legend("bottomleft", legend = c("Observations", "Means"), pch = c(1, 4))
```



```
## In these two examples the isotonic regression fails.
## For Gene 2 we can see an antitonic relationship between concentration and
## response in the data. An antitonic regression would make more sense here.
## For Gene 3 the relationship seems to take the shape of a parabola.
## Isotonic regression might not be suitable here.


## (ii)

apply(matrix(1:3), 1, FUN = function(x) {
  plot(replicates, VPA.Isotonic[x,],
       ylab = "Response", xlab = "Concentration", main = paste("Gene", x))
  points(concentrations, meansEx3[x, ], pch = 4)
  points(concentrations, pava(meansEx3[x,], weights, decreasing = TRUE),
         type = "l")
})
```

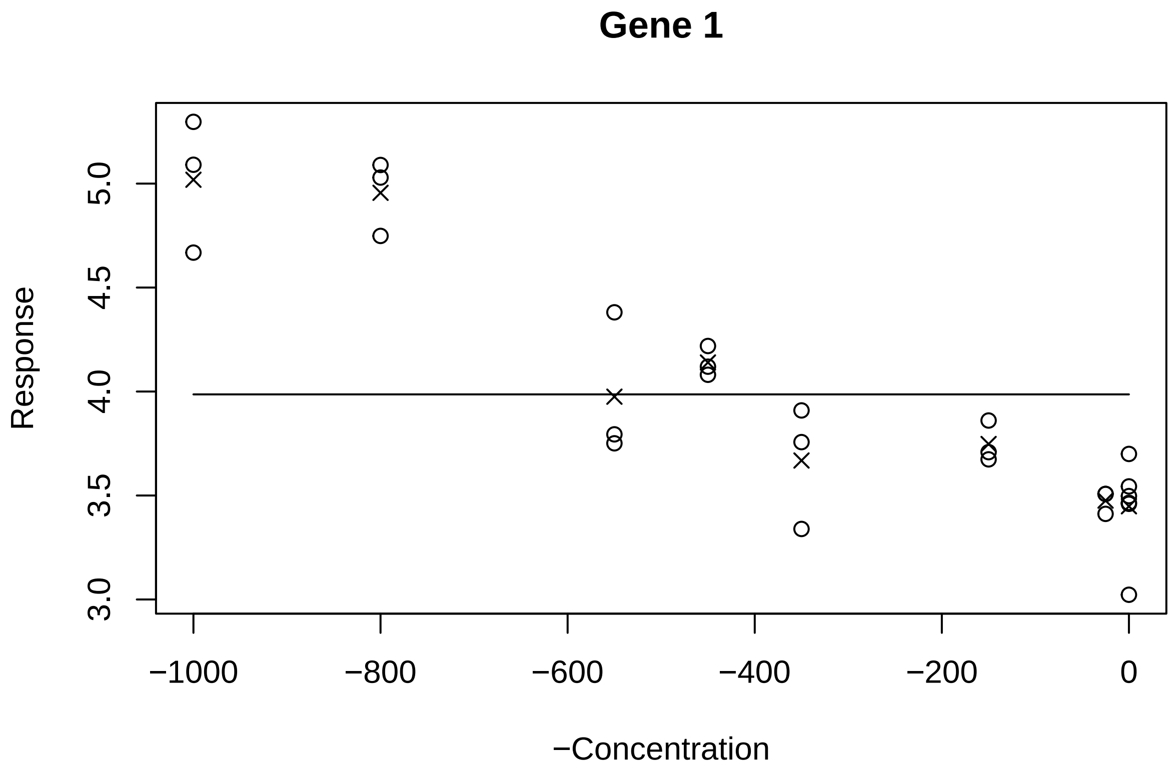

Gene 1
Response
3.0
3.5
4.0
4.5
5.0
0
200
400
600
800
1000
Concentration

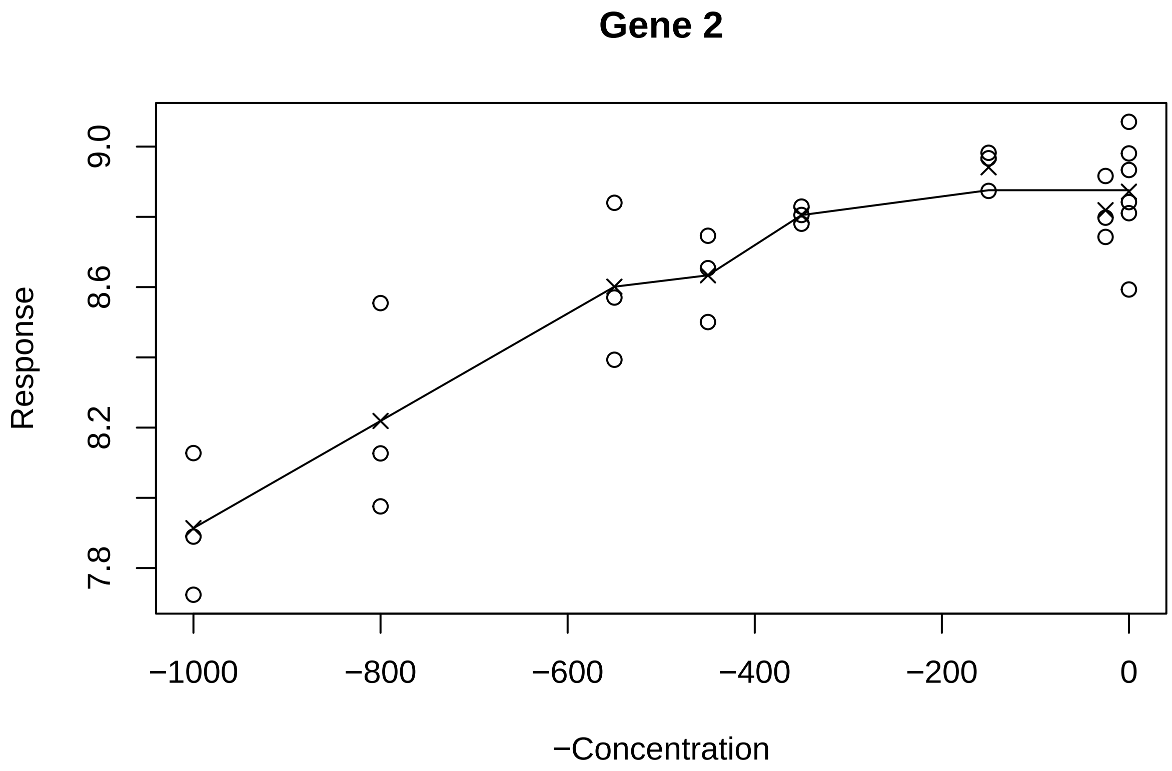

Gene 2
Response
9.0
8.6
8.2
7.8
0
200
400
600
800
1000
Concentration

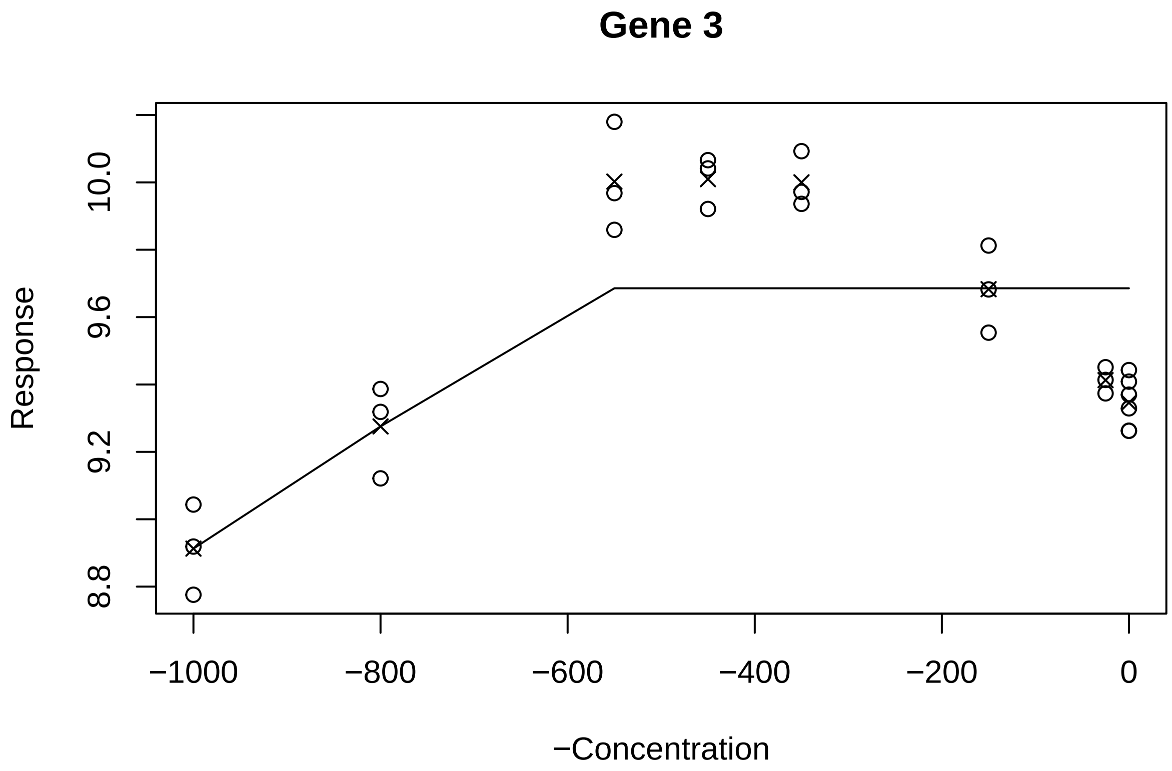


```
## NULL
```

```
## Antitonic regression fails for Gene 1 and 3 but works very well for Gene 2

## (iii)

## Reverse data by using -concentration:

apply(matrix(1:3), 1, FUN = function(x) {
  plot(-replicates, VPA.Isotonic[x,],
       ylab = "Response", xlab = "-Concentration", main = paste("Gene", x))
  points(-concentrations, meansEx3[x, ], pch = 4)
  points(-concentrations, rev(pava(rev(meansEx3[x,]), rev(weights))) ,
         type = "l")
})
```

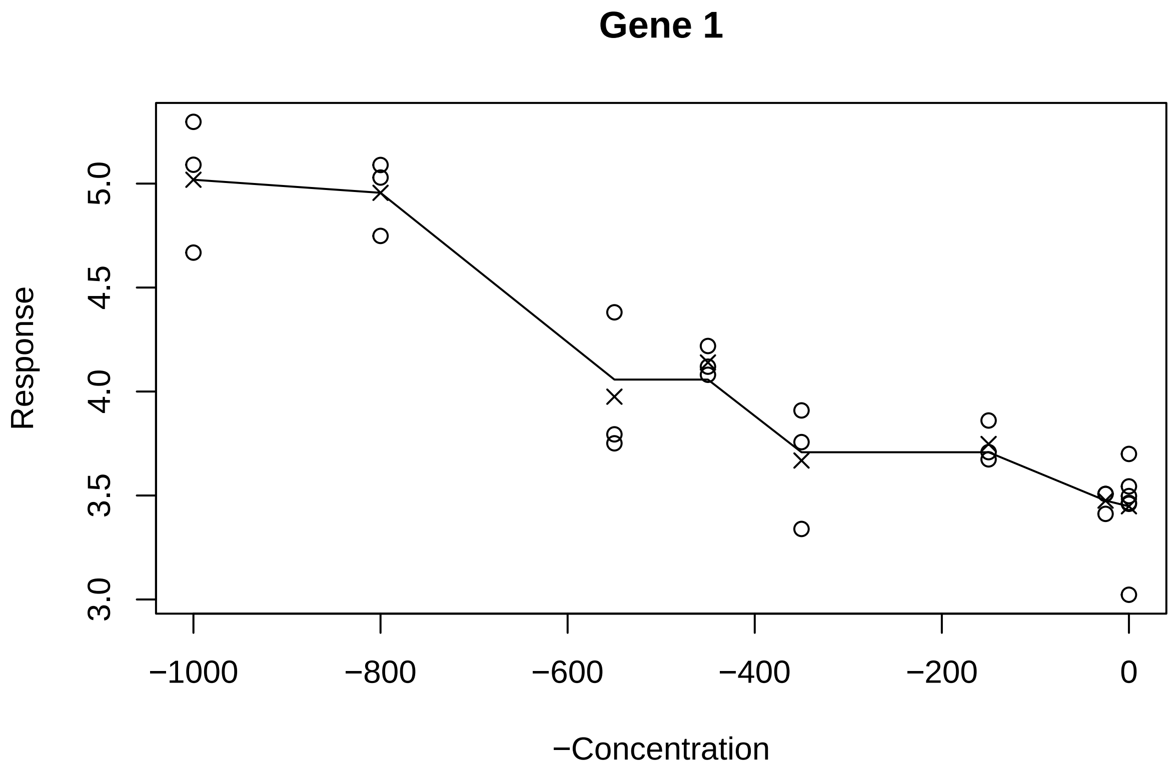

Gene 1
Response
5.0
4.5
4.0
3.5
3.0
−1000
−800
−600
−400
−200
0
−Concentration

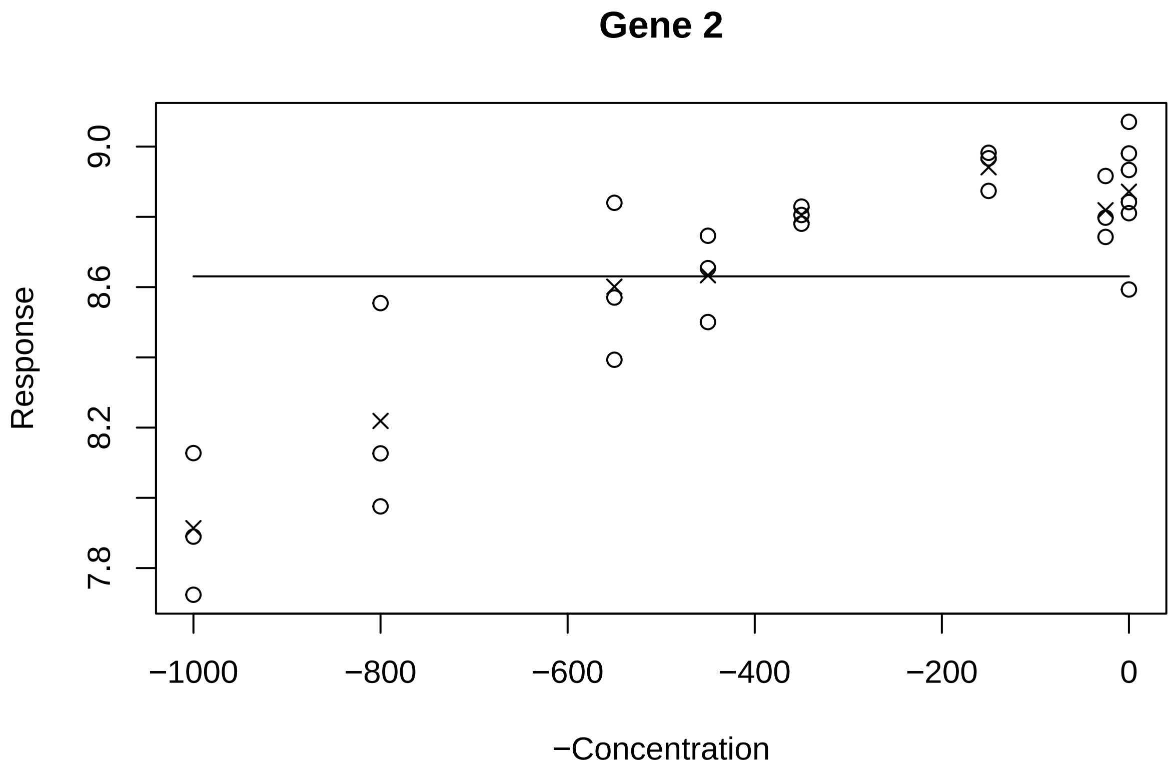

Gene 2
Response
9.0
8.6
8.2
7.8
−1000
−800
−600
−400
−200
0
−Concentration

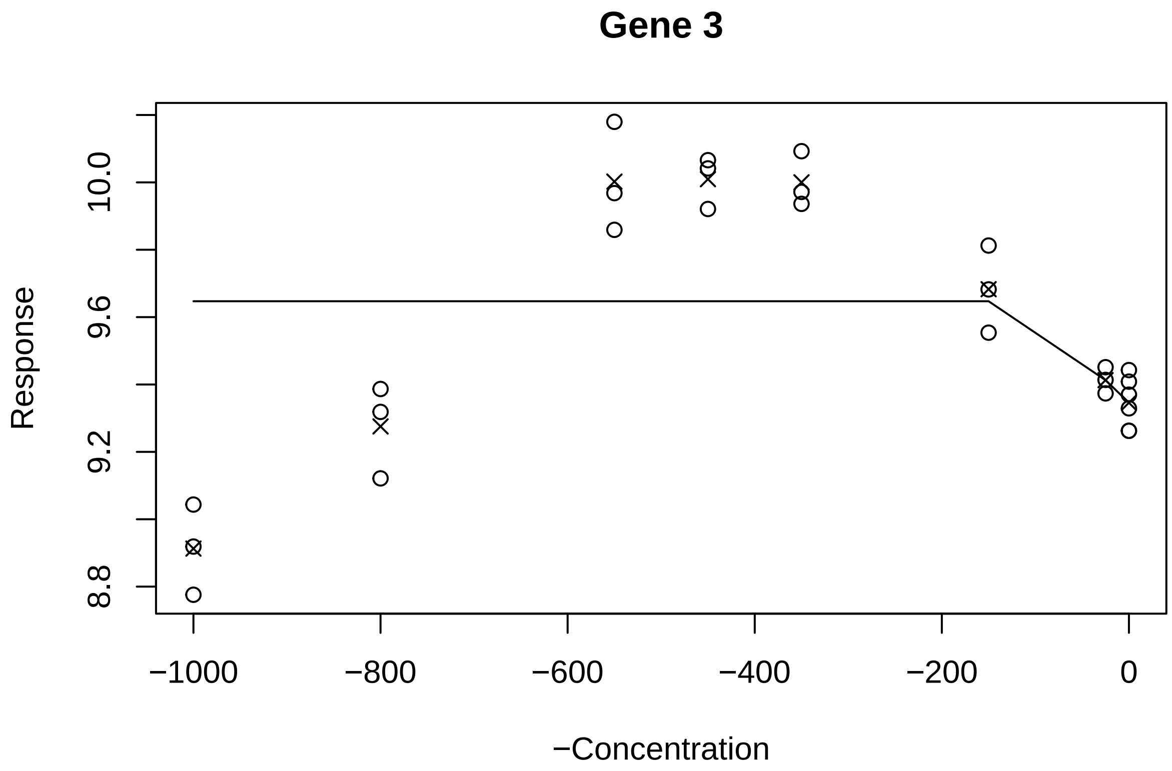


```
## NULL
```

```
## It has the same effect as performing antitonic regression on the original
## data

## (iv)

## Reverse data by using -concentration and performing antitonic regression:

apply(matrix(1:3), 1, FUN = function(x) {
  plot(-replicates, VPA.Isotonic[x,],
       ylab = "Response", xlab = "-Concentration", main = paste("Gene", x))
  points(-concentrations, meansEx3[x, ], pch = 4)
  points(-concentrations, rev(pava(rev(meansEx3[x,]), rev(weights),
                                   decreasing = TRUE)) ,
         type = "l")
})
```


Gene 1
Response
5.0
4.5
4.0
3.5
3.0
−1000
−800
−600
−400
−200
0
−Concentration


Gene 2
Response
9.0
8.6
8.2
7.8
−1000
−800
−600
−400
−200
0
−Concentration



```
## NULL
```

```
## As expected, it is equivalent to performing isotonic regression on original
## data.
```

Exercise 2 – PAVA I

a) PAVA – Algorithm:

1. Violation: Pool x_2 and x_3:

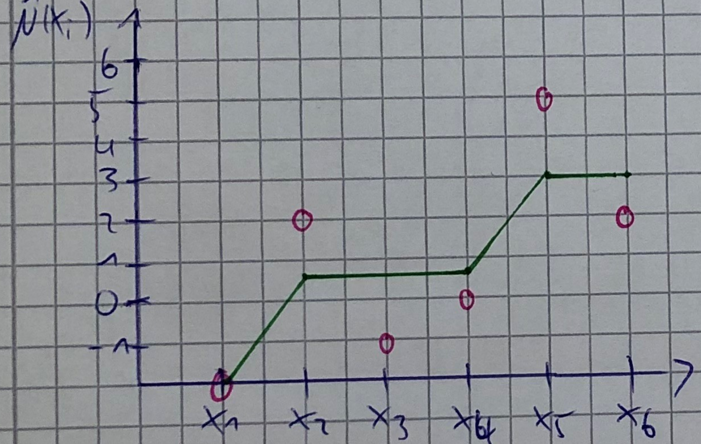
$$\hat{\mu}(x_2, x_3) = \frac{4\cdot 2 - 2\cdot 1}{4+2} = 1$$

=> Violation to x_4, so pool again:

$$\hat{\mu}(x_2, x_3, x_4) = \frac{6\cdot 1 + 4\cdot 0}{6+4} = \frac{6}{10} = 0.6$$

2. Violation: Pool x_5 and x_6:

$$\hat{\mu}(x_5, x_6) = \frac{4\cdot 5 + 8\cdot 2}{4+8} = \frac{36}{12} = 3$$

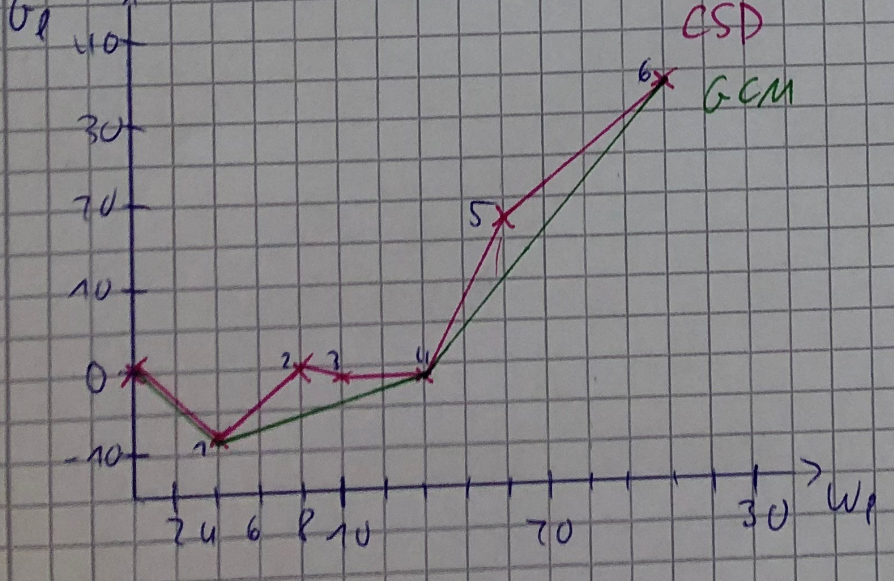


b)

$-4\cdot 2$ $+4\cdot 2$ $-2\cdot 1$ $+4\cdot 0$ $+4\cdot 5$ $+8\cdot 2$

l	0	1	2	3	4	5	6
G_l	0	-8	0	-2	-2	18	34
W_l	0	4	8	10	14	18	26

- No. of final sets: 3 (= No. of straight lines in GCM plot)
- The points for which CSD = GCM are pooled together with the next point for which CSD = GCM.


G_l
CSD
GCM
W_l