Solution - Sheet 1

Jaya Bharatam 235291 Agatha Anna Baby 235293 Amrutha Manoharan 236892
Nishant Madan 230370

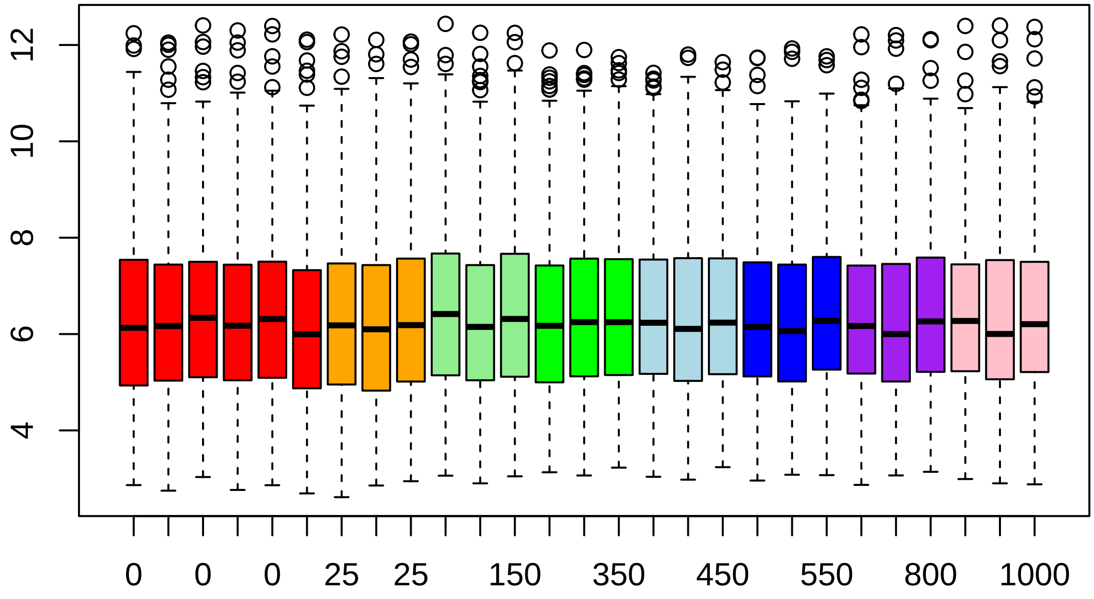
4/7/2022

Exercise 1: Descriptive analysis of the VPA dataset

```
library(ggplot2)
load("E:/TUD/04.Semester/toxicology-I/assignments/1/VPAData-Random.Rda")
concentration <-
  c(rep(0, 6), rep(c(25, 150, 350, 450, 550, 800, 1000), each = 3))
concentrations = rep(unique(concentration), each = ~~10000)~~
```

(a) Display the entire dataset via boxplots, analogously to the Figure from the lecture (Chapter 1.4, slide 22), i.e. with one boxplot per concentrations and replicate.

```
df <- data.frame(randomVPA)
colnames(df) <- concentration
colors = c(
  rep("red", 6),
  rep("orange", 3),
  rep("lightgreen", 3),
  rep("green", 3),
  rep("lightblue", 3),
  rep("blue", 3),
  rep("purple", 3),
  rep("pink", 3)
)
boxplot(df, col = colors)
```



✓

Each boxplot, shown above is almost the same variability. For each concentrations they have their maximum lies not above 12 and minimum not below 0. They are grouped and colored as for each concentrations.

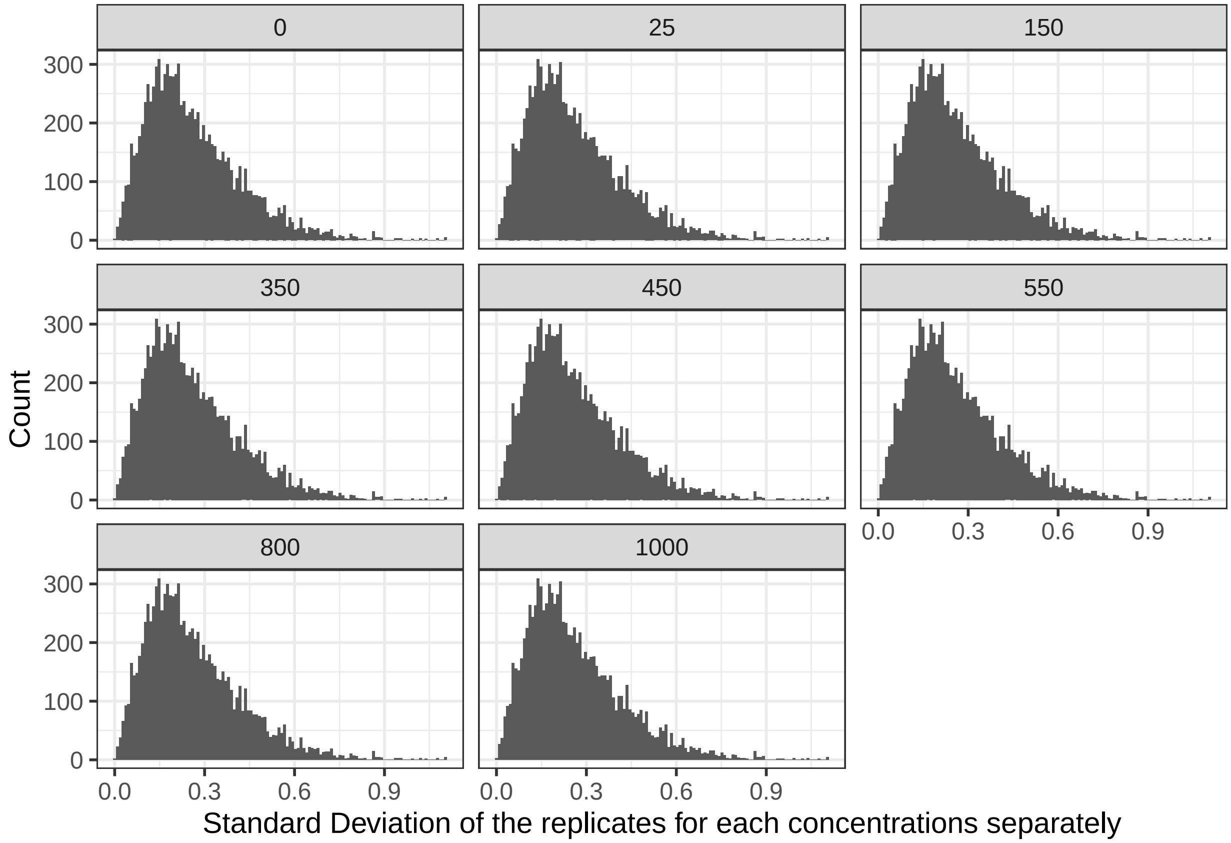
2/2

(b) For each gene, calculate the standard deviation of all three / six replicates corresponding to the same concentration This yields 8 values for each gene. Visualize the results stratified by concentration, i.e. summarize the results via a histogram for each concentration separately.

```
sd.by.concentrations <- t(apply(randomVPA, 1, function(x) {
  tapply(x, concentration, sd)
}))

sd.df <- data.frame(sd = c(sd.by.concentrations),
                    concentration = concentrations)

ggplot(sd.df, aes(x = sd)) +
  geom_histogram(bins = 120) +
  facet_wrap(. ~ concentration) +
  theme_bw() +
  xlab("Standard Deviation of the replicates for each concentrations separately") +
  ylab("Count")
```



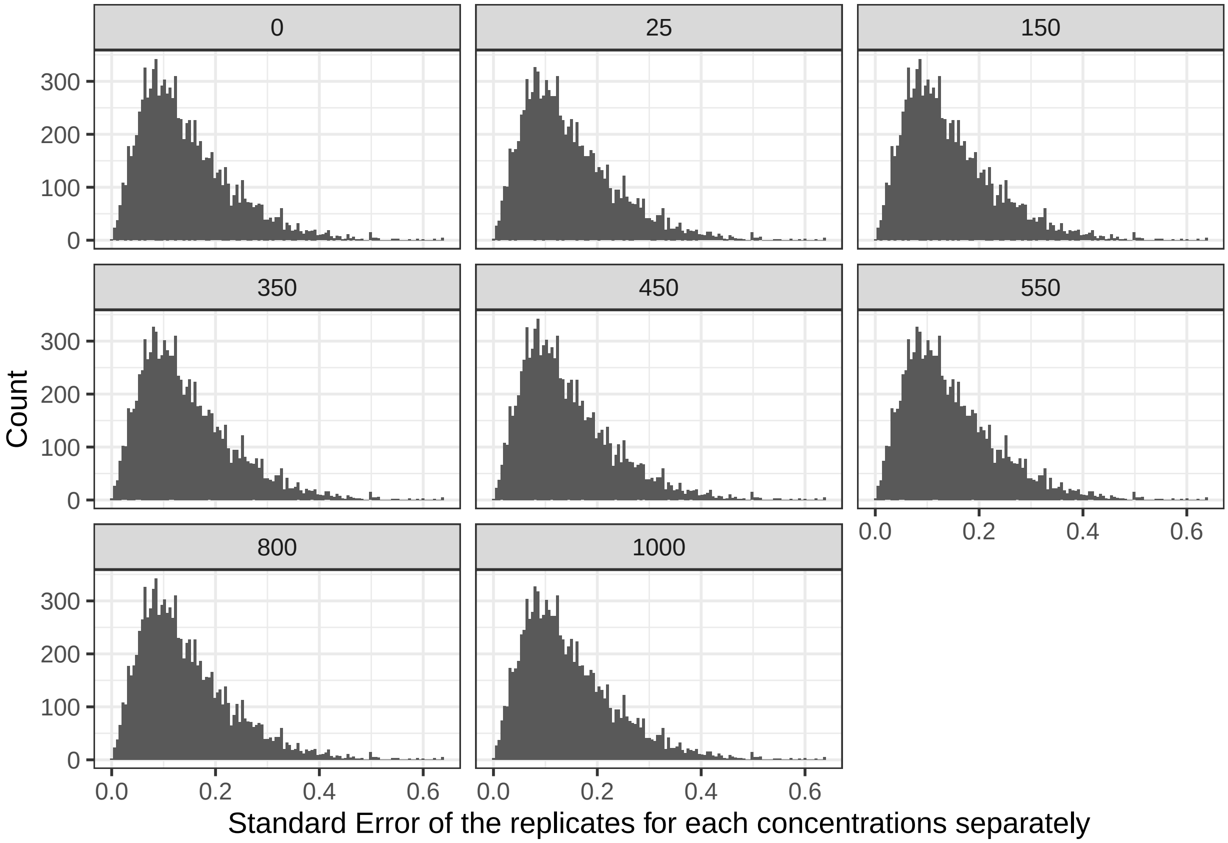
All the standard error histograms are positively right-skewed. Some of there peaks lies almost at 0.2, while some other have their maximum value at 0.7 and even more to 1 as well. ✓

1/2

(c) For each concentrations separately, find the gene for which the three / six replicates have the highest standard deviation. Plot the concentrations-response profiles for the resulting eight genes. Repeat the analysis, now finding the gene for which the three / six replicates have the lowest standard deviation, for each concentrations separately. Plot the concentrations-response profiles for the resulting eight genes.

```
se.by.concentration.nocontrol <-
  t(apply(randomVPA[,-(1:6)], 1, function(x) {
    tapply(x, concentration[-(1:6)], function(y)
      sd(y) / sqrt(3))
  }))
se.control <- apply(randomVPA[, 1:6], 1, function(y)
  sd(y) / sqrt(6))
se.by.concentration <-
  cbind(se.control, se.by.concentration.nocontrol)
se.df <- data.frame(se = c(se.by.concentration),
                    concentration = concentrations)
ggplot(se.df, aes(x = se)) +
  geom_histogram(bins = 120) +
  facet_wrap(. ~ concentration) +
  theme_bw() +
  xlab("Standard Error of the replicates for each concentrations separately") +
  ylab("Count")
```

this was last year's task...

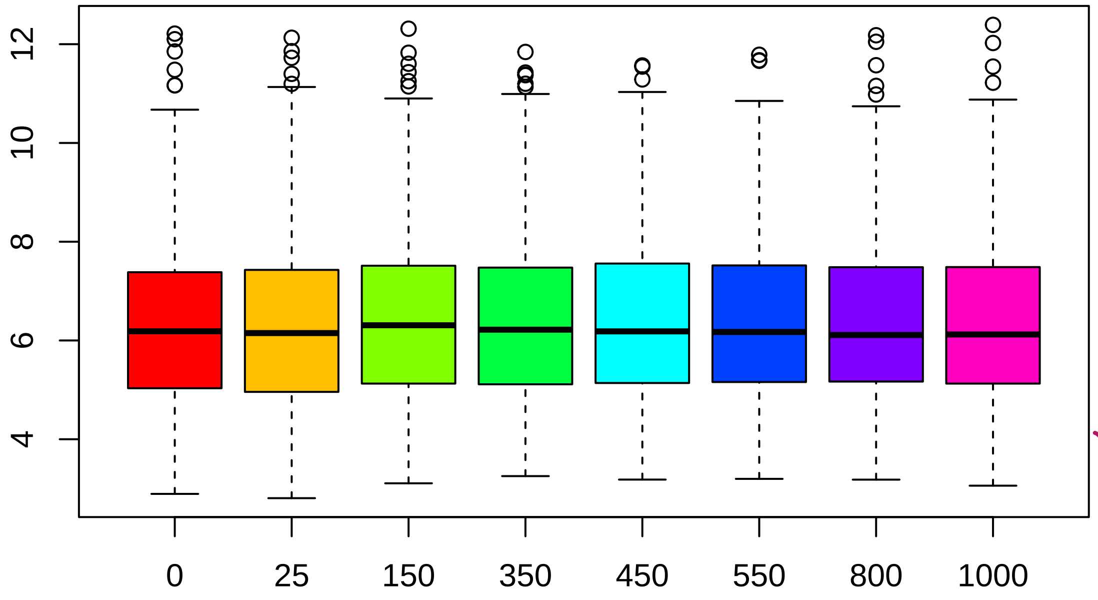


The results compare to b) are almost similar with a compression of the histograms for all positive concentrations by the factor 1/sqrt(3) nearly equal to 0.58. The histogram for the control is more compressed, by 1/sqrt(6) nearly equal to 0.41, and hence it becomes even more narrower.

0/4

(d) For each gene, calculate the mean expression value per concentrations This yields a reduced dataset with dimensions 500 and 8. Visualize the reduced dataset via boxplots analogously to (a), but now with only one plot per concentrations

```
mean.per.concentration <- t(apply(randomVPA, 1, function(x) {
  tapply(x, concentration, mean)
}))
boxplot(mean.per.concentration, col = rainbow(length(unique(concentration))))
```



2/2

Each concentration(mean.per.concentration) boxplot is nearly the same as the other. Showing above with different colors as for each concentrations respectively.

(e) Determine the number of genes for which the mean gene expression is monotonically increasing, and the number of genes for which it is monotonically decreasing. Plot the concentrations-response profiles for all genes with monotonically increasing and monotonically decreasing profiles. Make sure to indicate both the individual measurements per concentrations and the mean expression value per concentrations in the plot.

```
monoton.increasing <- apply(mean.per.concentration, 1, function(x) {
  all(diff(x) > 0)
})
sum(monoton.increasing)
```

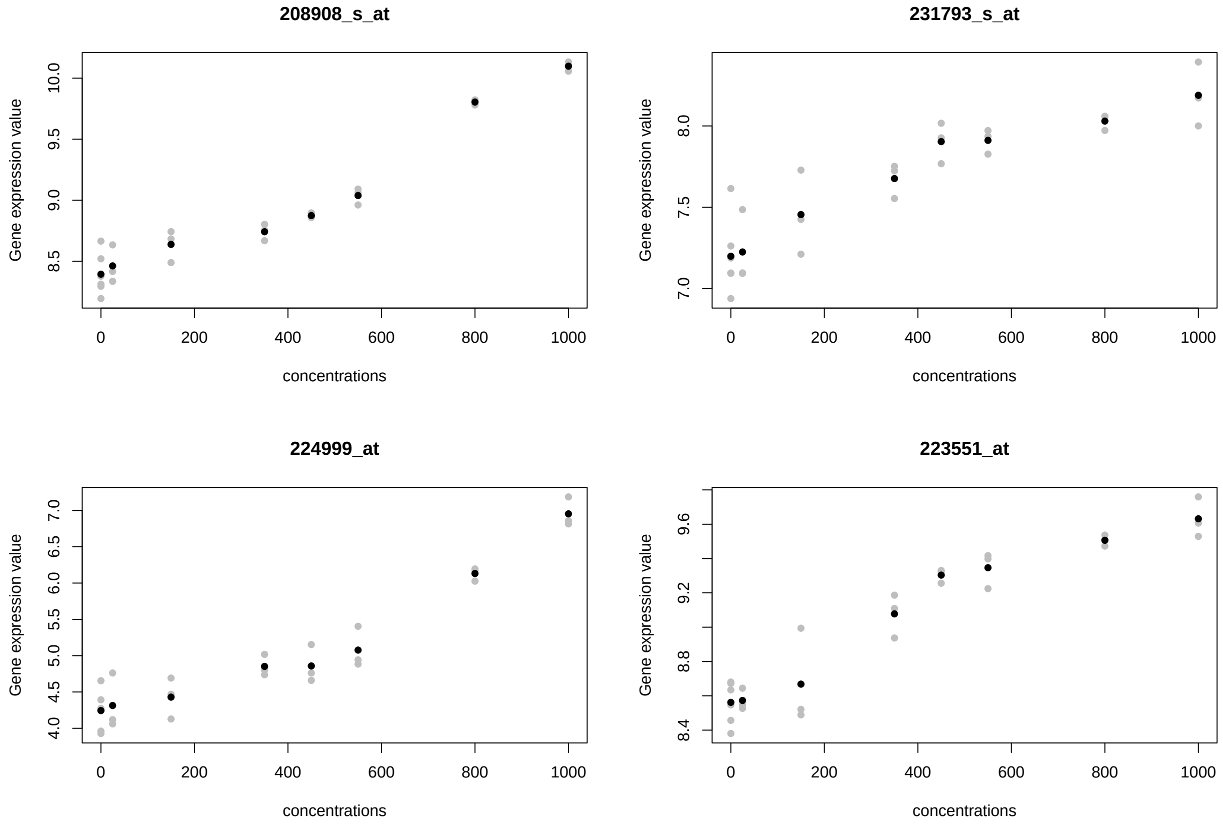
```
## [1] 19
```

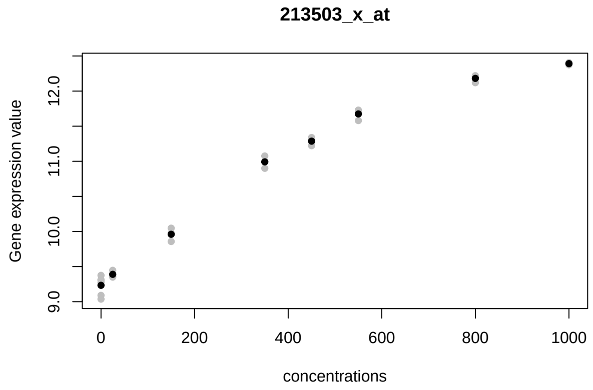
```
monoton.decreasing <- apply(mean.per.concentration, 1, function(x) {
  all(c(x[1] > x[2], x[2] > x[3], x[3] > x[4], x[4] > x[5], x[5] > x[6], x[6] >
          x[7], x[7] > x[8]))
})
sum(monoton.decreasing)
```

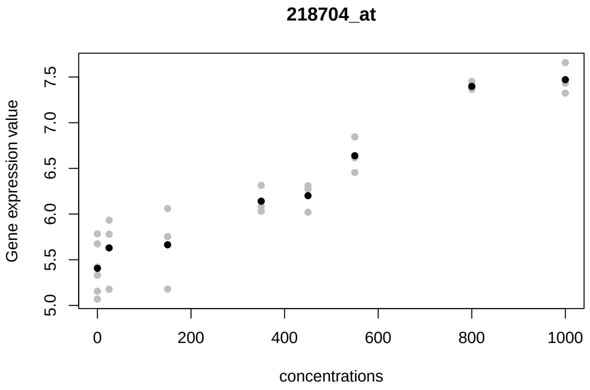
```
## [1] 15
```

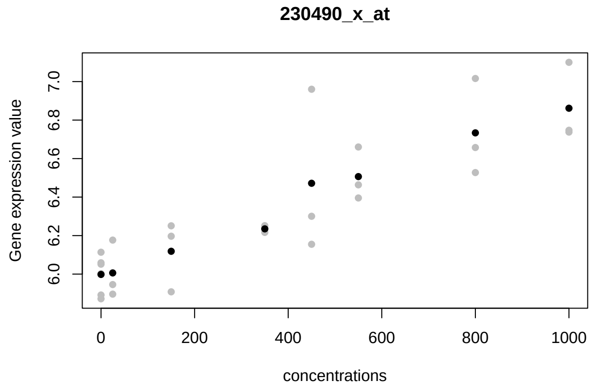
The number of genes with monotonically increasing expression values (19) and the number of genes with monotonically decreasing expression values (15) is similar and rather low. Thus, the vast majority of genes do not exhibit a monotonic profile.

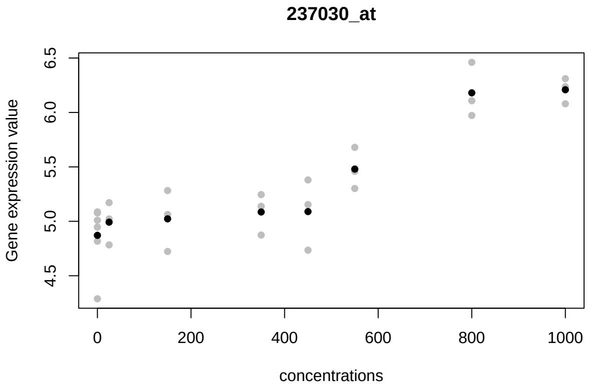
```
inc.mon <- which(monoton.increasing == TRUE)
for (i in inc.mon) {
  plot(
    concentration,
    randomVPA[i, ],
    pch = 16,
    col = "grey",
    xlab = "concentrations",
    ylab = "Gene expression value",
    main = rownames(randomVPA)[i]
  )
  points(unique(concentration),
         mean.per.concentration[i, ],
         pch = 16,
         col = "black")
}
```

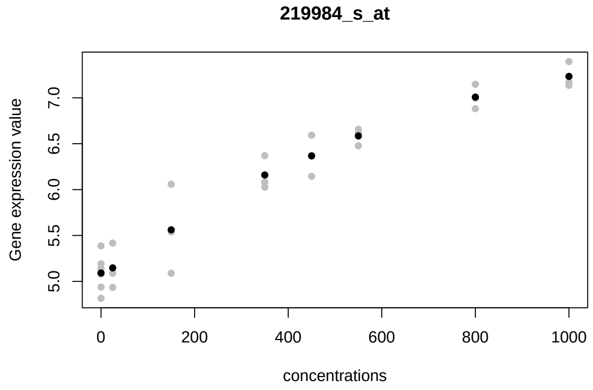


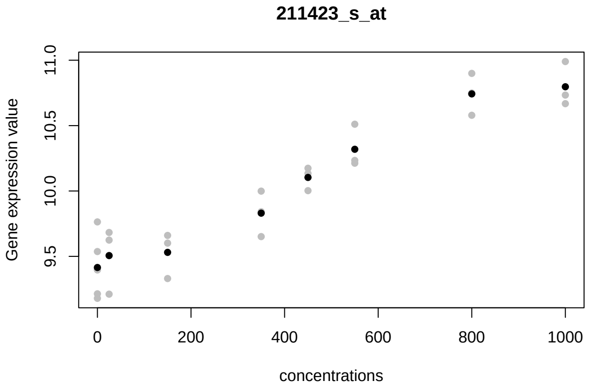

213503_x_at
Gene expression value
9.0
10.0
11.0
12.0
0
200
400
600
800
1000
concentrations

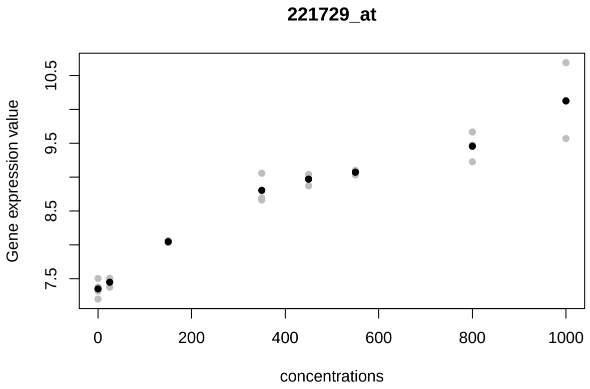

218704_at
Gene expression value
5.0
5.5
6.0
6.5
7.0
7.5
0
200
400
600
800
1000
concentrations


230490_x_at
Gene expression value
6.0
6.2
6.4
6.6
6.8
7.0
0
200
400
600
800
1000
concentrations

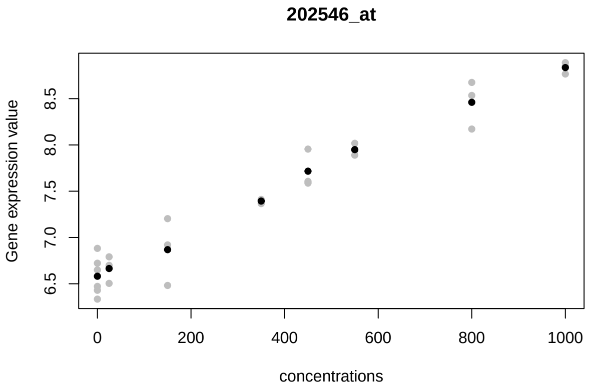

237030_at
Gene expression value
4.5
5.0
5.5
6.0
6.5
0
200
400
600
800
1000
concentrations


219984_s_at
Gene expression value
5.0
5.5
6.0
6.5
7.0
0
200
400
600
800
1000
concentrations

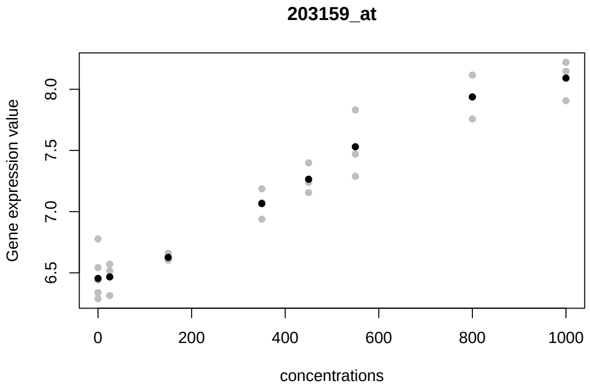

211423_s_at
Gene expression value
9.5
10.0
10.5
11.0
0
200
400
600
800
1000
concentrations



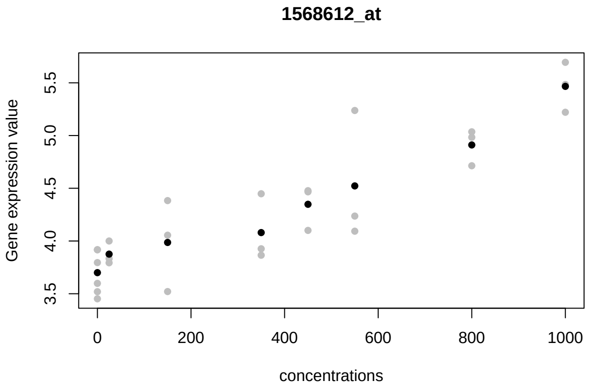
221729_at



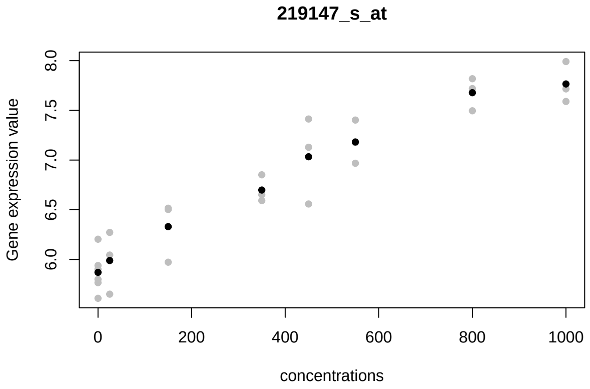
202546_at



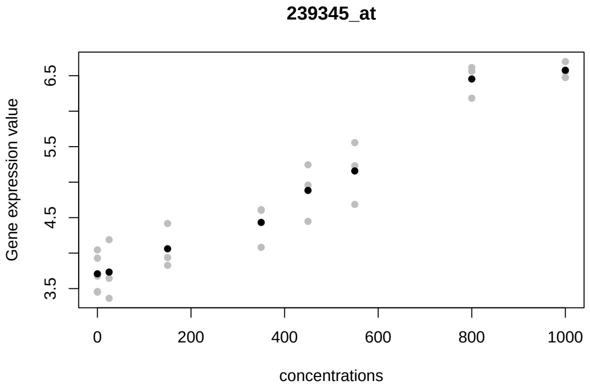
203159_at



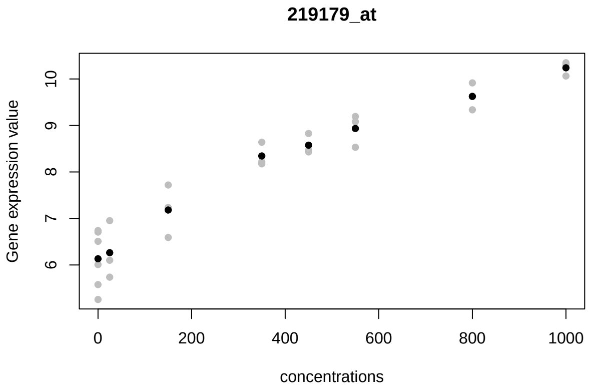
1568612_at

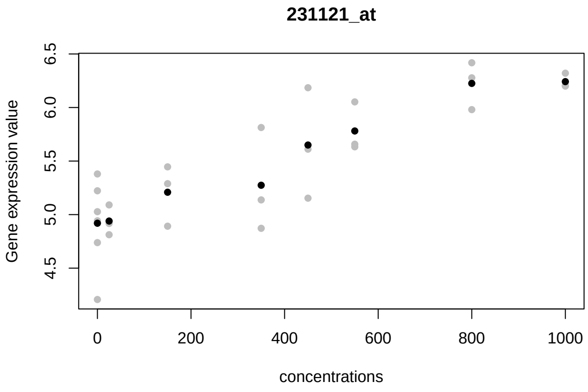


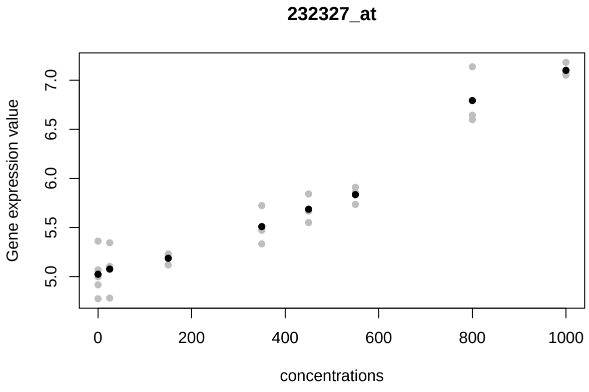
219147_s_at



239345_at

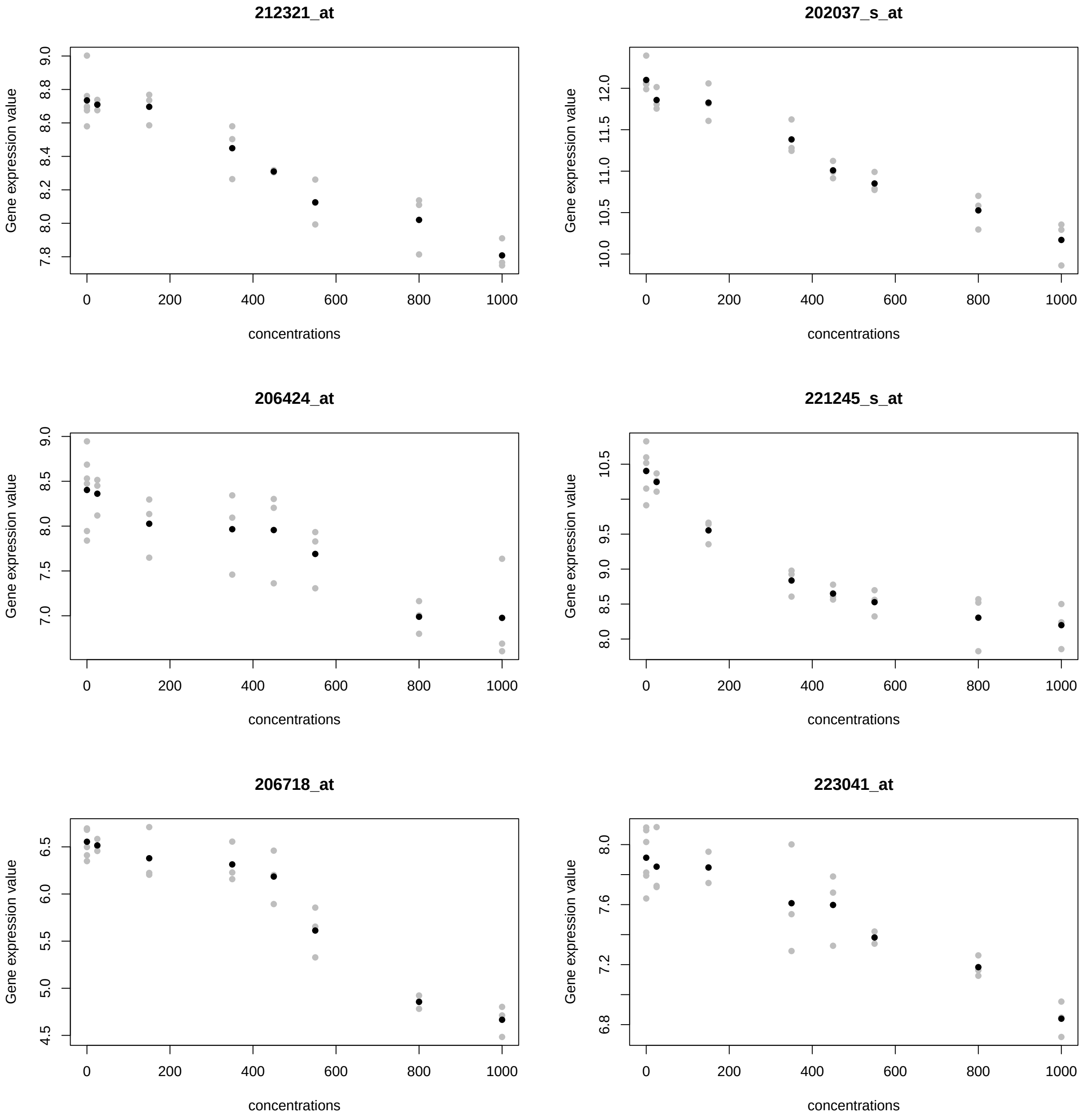


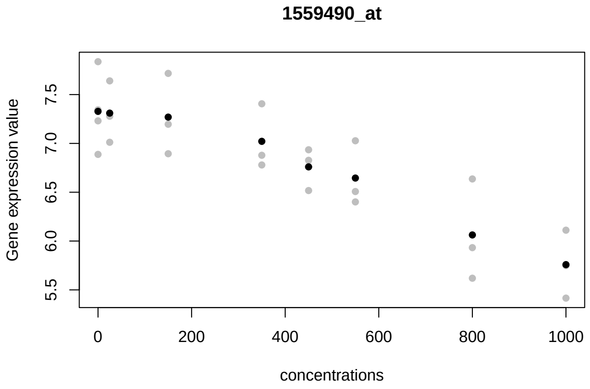


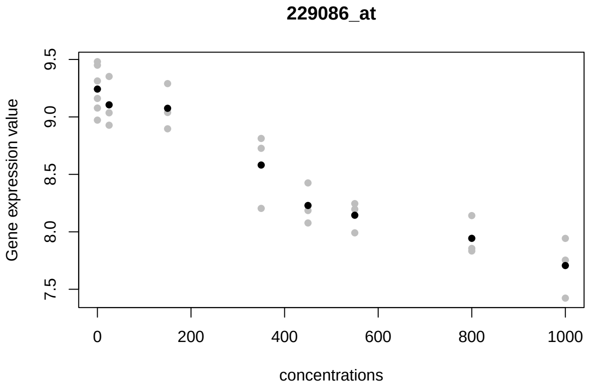


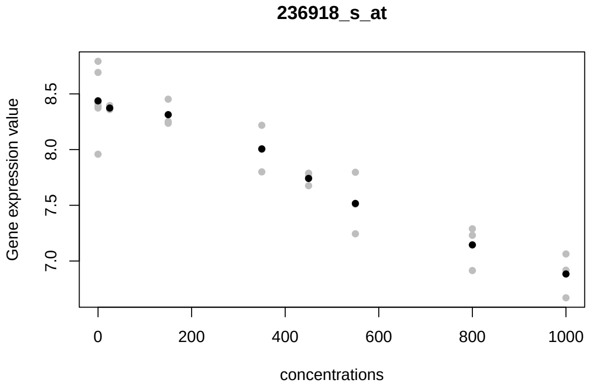
✓

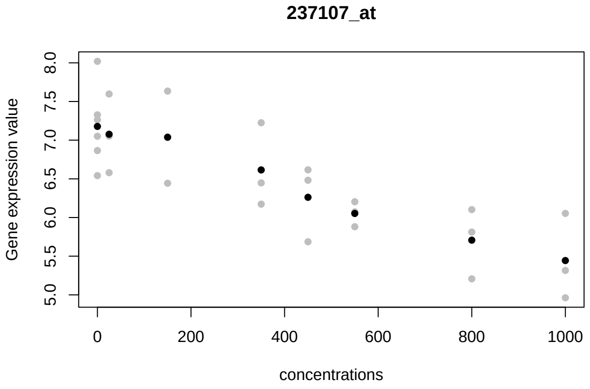
```r
dec.mon <- which(monoton.decreasing == TRUE)
for (i in dec.mon) {
  plot(
    concentration,
    randomVPA[i, ],
    pch = 16,
    col = "grey",
    xlab = "concentrations",
    ylab = "Gene expression value",
    main = rownames(randomVPA)[i]
  )
  points(unique(concentration),
         mean.per.concentration[i, ],
         pch = 16,
         col = "black")
}
```

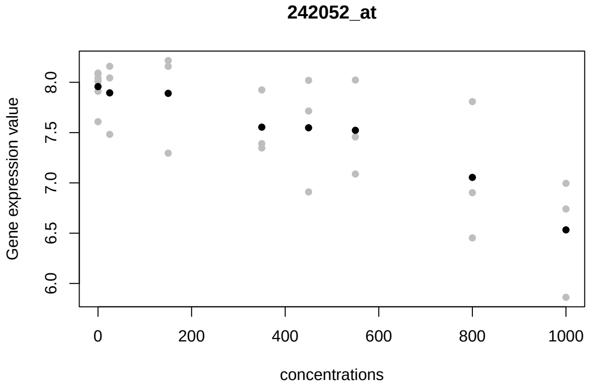

212321_at
Gene expression value
7.8
8.0
8.2
8.4
8.6
8.8
9.0
0
200
400
600
800
1000
concentrations
202037_s_at
Gene expression value
10.0
10.5
11.0
11.5
12.0
0
200
400
600
800
1000
concentrations
206424_at
Gene expression value
7.0
7.5
8.0
8.5
9.0
0
200
400
600
800
1000
concentrations
221245_s_at
Gene expression value
8.0
8.5
9.0
9.5
10.5
0
200
400
600
800
1000
concentrations
206718_at
Gene expression value
4.5
5.0
5.5
6.0
6.5
0
200
400
600
800
1000
concentrations
223041_at
Gene expression value
6.8
7.2
7.6
8.0
0
200
400
600
800
1000
concentrations

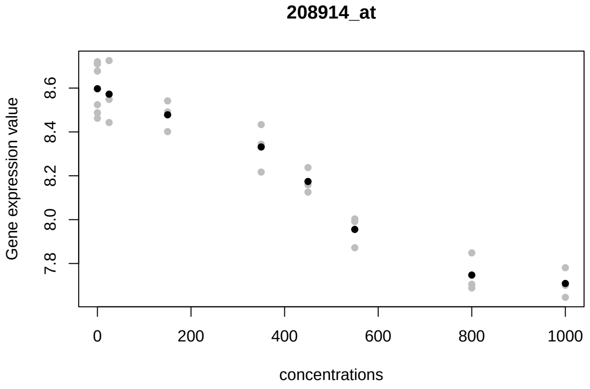

1559490_at
Gene expression value
5.5
6.0
6.5
7.0
7.5
0
200
400
600
800
1000
concentrations

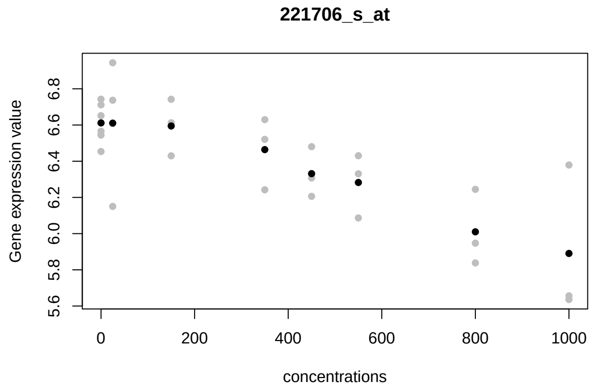

229086_at
Gene expression value
7.5
8.0
8.5
9.0
9.5
0
200
400
600
800
1000
concentrations

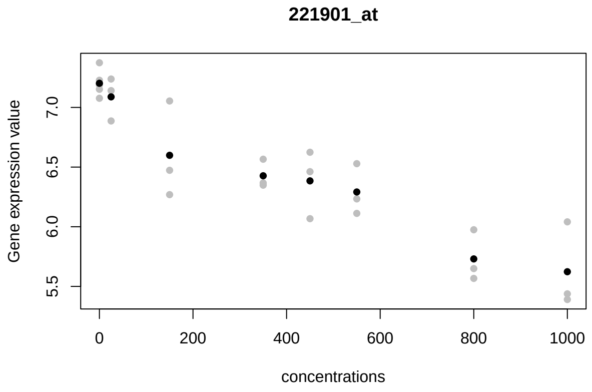

236918_s_at
Gene expression value
7.0
7.5
8.0
8.5
0
200
400
600
800
1000
concentrations

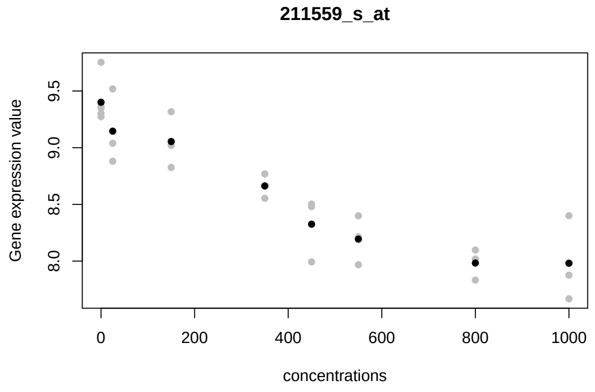

237107_at
Gene expression value
5.0
5.5
6.0
6.5
7.0
7.5
8.0
0
200
400
600
800
1000
concentrations


242052_at
Gene expression value
6.0
6.5
7.0
7.5
8.0
0
200
400
600
800
1000
concentrations


208914_at
Gene expression value
7.8
8.0
8.2
8.4
8.6
0
200
400
600
800
1000
concentrations





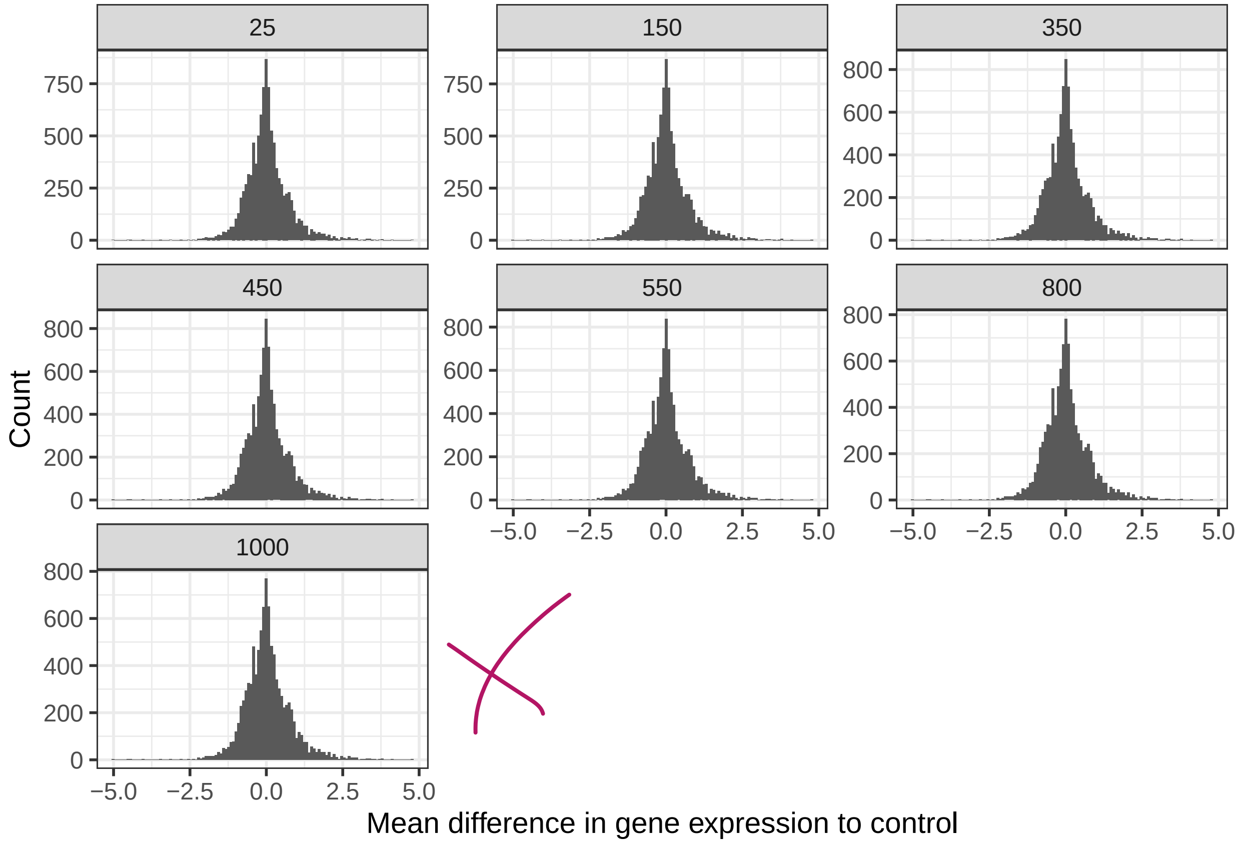


✓

3/3

(f) For each gene, calculate the difference between the mean gene expression value of each positive concentration and the mean gene expression value of the control. Visualize the results with one histogram per concentration.

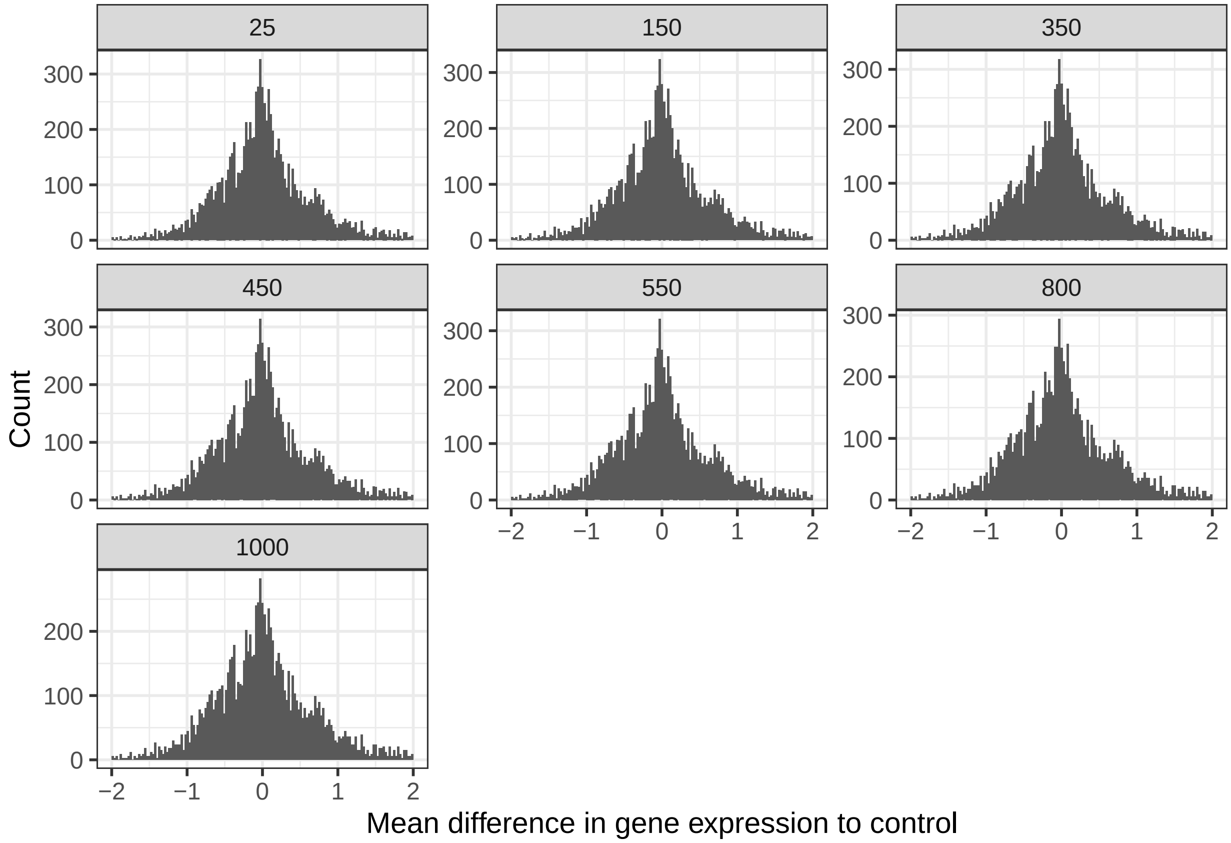
```
f.change <- t(apply(mean.per.concentration, 1, function(x) {
  x[2:8] - x[1]
}))
f.change.df <- data.frame(
  FC = c(
    f.change[, 1],
    f.change[, 2],
    f.change[, 3],
    f.change[, 4],
    f.change[, 5],
    f.change[, 6],
    f.change[, 7]
  ),
  concentration = rep(unique(concentration)[-1], each = 10000)
)
ggplot(f.change.df, aes(x = FC)) +
  geom_histogram(bins = 120) +
  facet_wrap(. ~ concentration, scales = "free_y") +
  theme_bw() +
  xlab("Mean difference in gene expression to control") +
  ylab("Count")
```



```
ggplot(f.change.df, aes(x = FC)) +
  geom_histogram(bins = 150) +
  facet_wrap(. ~ concentration, scales = "free_y") +
  theme_bw() +
  xlab("Mean difference in gene expression to control") +
  ylab("Count") + xlim(-2, 2)
```

```
## Warning: Removed 1680 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 7 rows containing missing values (geom_bar).
```



Major differences is in the range between -2 and 2. With increasing concentrations, the histograms become slightly broader, proves larger differences in expression values (“fold changes”). Additionally, indicates that here only very few f.change of approximately 0 are observed.

↳ no

↳ no

0/2

Σ 8/15