# HW2

113078506

2025-02-28

## Question 1

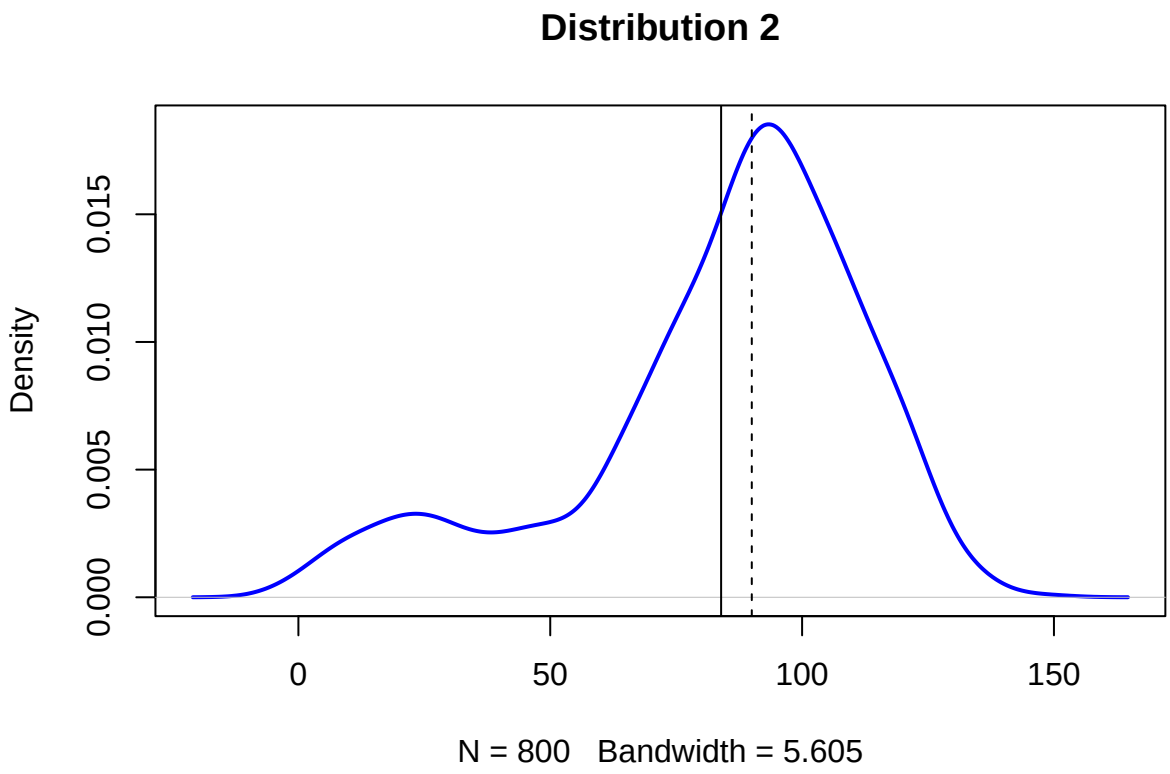
### (a)

```r
# Three normally distributed data sets
d1 <- rnorm(n=500, mean=100, sd=15)
d2 <- rnorm(n=200, mean=75, sd=15)
d3 <- rnorm(n=100, mean=25, sd=15)

# Combining them into a composite dataset
d123 <- c(d1, d2, d3)

# Let's plot the density function of d123
plot(density(d123), col="blue", lwd=2,
     main = "Distribution 2")

# Add vertical lines showing mean and median
abline(v=mean(d123))
abline(v=median(d123), lty="dashed")
```

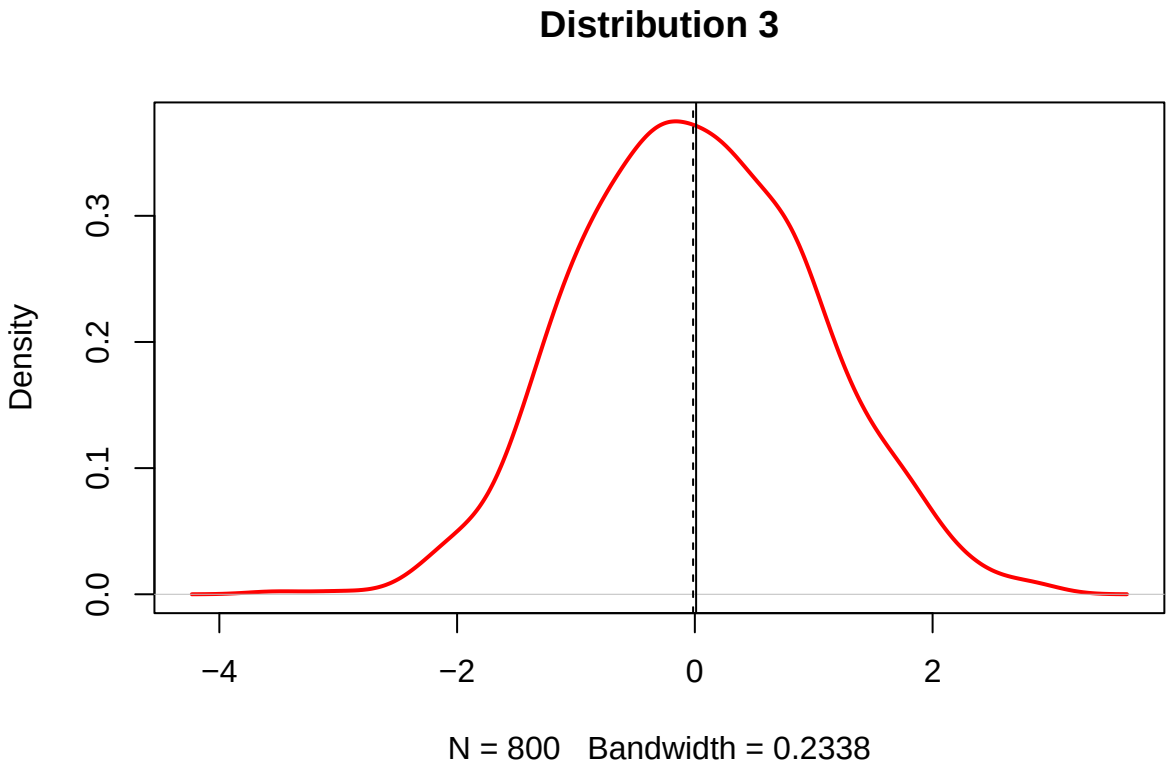



(b)

```r
dataset = rnorm(n=800)

plot(density(dataset), col="red", lwd=2,
     main = "Distribution 3")

abline(v=mean(dataset))
abline(v=median(dataset), lty="dashed")
```




(c)

- Mean will be more sensitive when outliers are added to the dataset, because mean is calculated by summing the whole data and divided by the length, which is affected easily by extreme value, whereas median is a middle value from sorted dataset sequence, which is not easily affected by the extreme value.

## Question 2

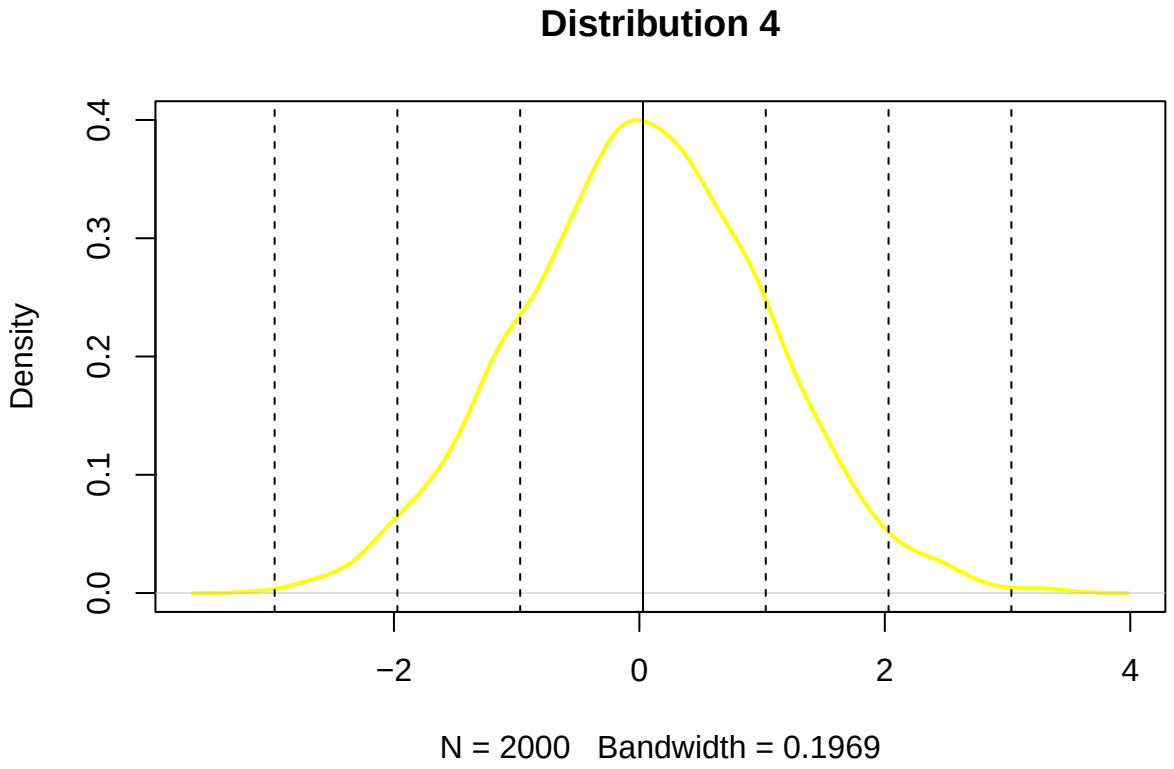
(a)

```
set.seed(123)
rdata = rnorm(n=2000, mean = 0, sd = 1)

plot(density(rdata), col="yellow", lwd=2,
     main = "Distribution 4")

abline(v=mean(rdata))
abline(v=mean(rdata)+sd(rdata), lty="dashed")
abline(v=mean(rdata)+sd(rdata)*2, lty="dashed")
abline(v=mean(rdata)+sd(rdata)*3, lty="dashed")
abline(v=mean(rdata)-sd(rdata), lty="dashed")
```

```
abline(v=mean(rdata)-sd(rdata)*2, lty="dashed")
abline(v=mean(rdata)-sd(rdata)*3, lty="dashed")
```




(b)

```
Q1 <- quantile(rdata, 1/4)
Q2 <- quantile(rdata, 2/4)
Q3 <- quantile(rdata, 3/4)
quan_data <- c(Q1, Q2, Q3)
quan_data
```

```
##         25%         50%         75%
## -0.64395403  0.02881311  0.70859175
```

```
dist <- c((Q1-0)/1,(Q2-0)/1,(Q3-0)/1)
dist
```

```
##         25%         50%         75%
## -0.64395403  0.02881311  0.70859175
```

**(c)**

```
set.seed(123)
rdata_1 = rnorm(n=2000, mean=35, sd=3.5)

Q1 <- quantile(rdata_1, 1/4)
Q2 <- quantile(rdata_1, 2/4)
Q3 <- quantile(rdata_1, 3/4)
quan_data_1 <- c(Q1, Q2, Q3)
quan_data_1
```

```
##      25%      50%      75%
## 32.74616 35.10085 37.48007
```

```
dist_1 <- c((Q1-35)/3.5,(Q3-35)/3.5)
dist_1
```

```
##        25%        75%
## -0.6439540  0.7085917
```

- The result of (c) is almost the same to (b), Q1 both at approximately -0.64 sd, Q3 both at approximately 0.7.

**(d)**

```
Q1 <- quantile(d123, 1/4)
Q2 <- quantile(d123, 2/4)
Q3 <- quantile(d123, 3/4)
quan_data_2 <- c(Q1, Q2, Q3)
quan_data_2
```

```
##       25%       50%       75%
##  72.09931  90.01141 103.87197
```

```
dist_2 <- c((Q1-mean(d123))/sd(d123),(Q3-mean(d123))/sd(d123) )
dist_2
```

```
##        25%        75%
## -0.4072396  0.6873279
```

- The results of (d) is clearly different from (b), the absolute value of Q1 from (d) is significantly smaller than (b), and the Q3 from (d) is slightly smaller than Q3 from (b). The changes between the comparison might caused by negatively-skewed distribution from d123.

## Question 3

**(a)**

- Freedman–Diaconis rule: 2 * IQR(x) / length(x)^(1/3). According to WikiPedia, this formula can approximately minimize the integral of the squared difference between the histogram.

**(b)**

```r
set.seed(123)
rand_data <- rnorm(800, mean=20, sd = 5)

#Sturges's Formula
k_sturges = ceiling(log2(length(rand_data)))
h_sturges = (max(rand_data) - min(rand_data)) / k_sturges

cat("Sturges: k =", k_sturges, ", h =", h_sturges, "\n")
```

```
## Sturges: k = 10 , h = 3.025407
```

```r
#Scott's normal reference rule
h_scott <- (3.49 * sd(rand_data)) / (length(rand_data)^(1/3))
k_scott = ceiling((max(rand_data) - min(rand_data))/h_scott)

cat("Scott: k =", k_scott, ", h =", h_scott, "\n")
```

```
## Scott: k = 17 , h = 1.844283
```

```r
#Freedman-Diaconis' choice
h_fd <- (2 * IQR(rand_data)) / (length(rand_data)^(1/3))
k_fd <- ceiling((max(rand_data) - min(rand_data) / h_fd))

cat("Freedman-Diaconis: k =", k_fd, ", h =", h_fd, "\n")
```

```
## Freedman-Diaconis: k = 32 , h = 1.372593
```

**(c)**

```r
set.seed(123)
out_data <- c(rand_data, runif(10, min=40, max=60))

#Sturges's Formula
k_sturges = ceiling(log2(length(out_data)))
h_sturges = (max(out_data) - min(out_data)) / k_sturges

cat("Sturges: k =", k_sturges, ", h =", h_sturges, "\n")
```

```
## Sturges: k = 10 , h = 5.285822
```

```r
#Scott's normal reference rule
h_scott <- (3.49 * sd(out_data)) / (length(out_data)^(1/3))
k_scott = ceiling((max(out_data) - min(out_data))/h_scott)

cat("Scott: k =", k_scott, ", h =", h_scott, "\n")
```

```
## Scott: k = 24 , h = 2.254987
```

```r
#Freedman-Diaconis' choice
h_fd <- (2 * IQR(out_data)) / (length(out_data)^(1/3))
k_fd <- ceiling((max(out_data) - min(out_data) / h_fd))

cat("Freedman-Diaconis: k =", k_fd, ", h =", h_fd, "\n")
```

```
## Freedman-Diaconis: k = 55 , h = 1.394567
```

- The width calculated by Freedman-Diaconis' choice is least affected by the outliers, because it's calculated based on IQR, not the whole data range.