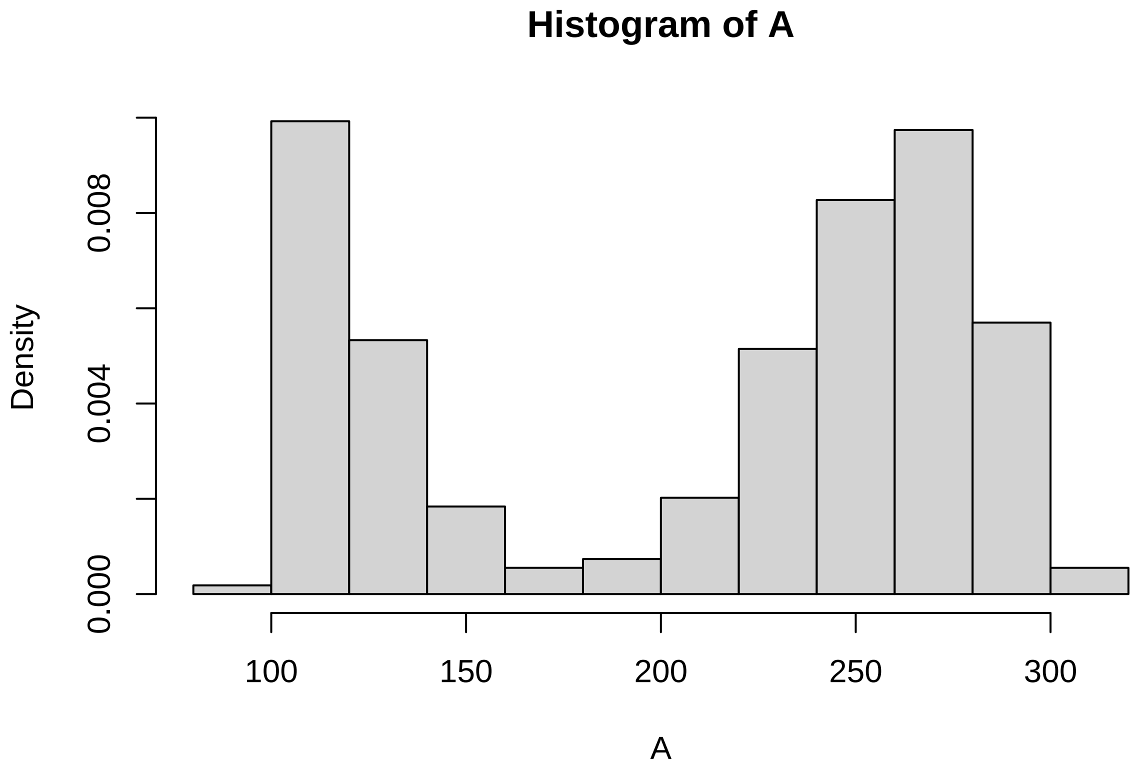
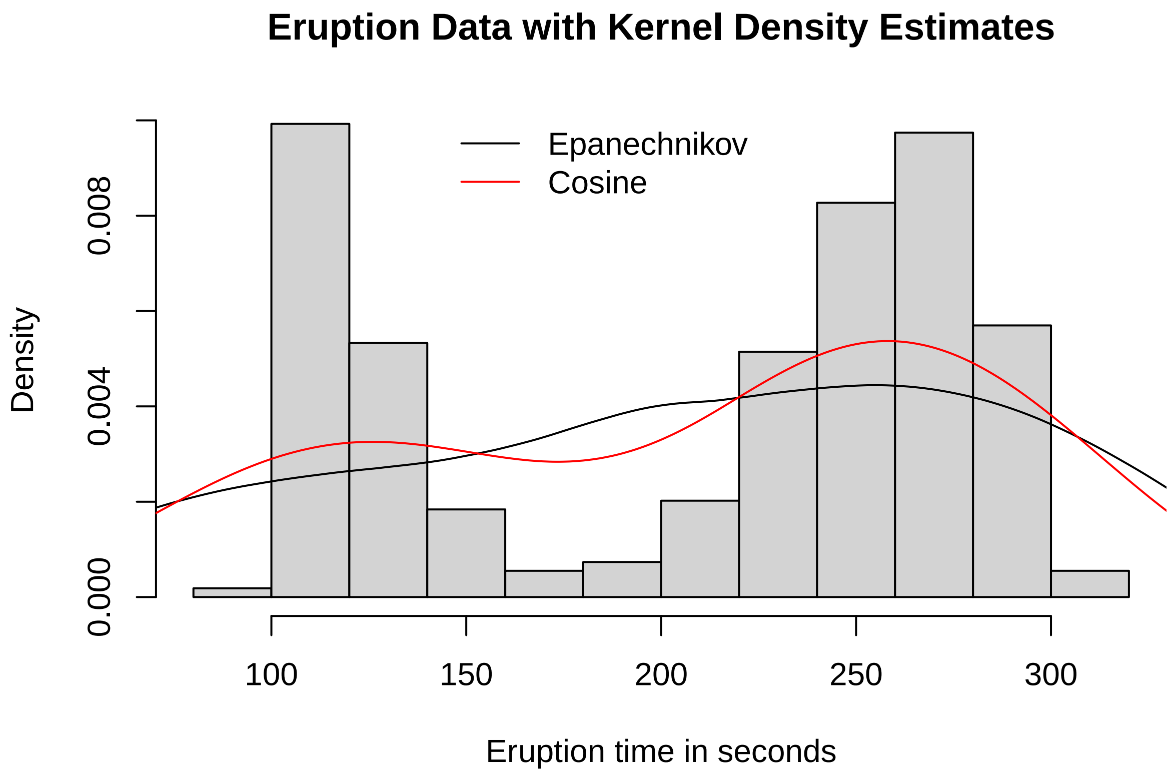
R Notebook

```
#Q.3(b)
data(faithful)

A <- faithful$eruptions * 60
hist(A,prob=T)
```



```
#Q.3(d)
s <- sd(A)
n <- length(A)

b_epanechnikov <- (1 / (n * (1/5)^2))^(1/5) * s * (8 / (3 * sqrt(pi)))^(1/5)
b_cosine <- (pi^2 / (16 * n * (1/4)^2))^(1/5) * s * (8 / (3 * sqrt(pi)))^(1/5)

hist(A, prob = TRUE, main = "Eruption Data with Kernel Density Estimates",
     xlab = "Eruption time in seconds")
lines(density(A, bw = b_epanechnikov, kernel = "epanechnikov"))
lines(density(A, bw = b_cosine, kernel = "cosine"), col = "red")
```

```r
legend(x = "topright", inset = c(0.4, 0), legend = c("Epanechnikov", "Cosine"),
       col = c("black", "red"), lty = 1, bty = "n")
```



```r
#Exercice 4:
  #(a)
  # Loading dataset
  data(WWWusage)

# quantiles
quantiles <- quantile(WWWusage)

# maximum and minimum
max_value <- max(WWWusage)
min_value <- min(WWWusage)

# mean and median
mean_value <- mean(WWWusage)
median_value <- median(WWWusage)

# IQR

q25 <- quantile(WWWusage, 0.25)
q75 <- quantile(WWWusage, 0.75)

iqr <- IQR(WWWusage)
```

```r
iqr_value <- IQR(WWWusage)

# Function for mode
calculate_mode <- function(x) {
  unique_x <- unique(x)
  unique_x[which.max(tabulate(match(x, unique_x)))]
}

mode_value <- calculate_mode(WWWusage)

# results
cat("Quantiles:\n")
```

```
## Quantiles:
```

```r
print(quantiles)
```

```
##    0%   25%   50%   75%  100% 
##  83.0  99.0 138.5 167.5 228.0
```

```r
cat("\nMaximum:", max_value)
```

```
## 
## Maximum: 228
```

```r
cat("\nMinimum:", min_value)
```

```
## 
## Minimum: 83
```

```r
cat("\nMean:", mean_value)
```

```
## 
## Mean: 137.08
```

```r
cat("\nMedian:", median_value)
```

```
## 
## Median: 138.5
```

```r
cat("\nIQR:", iqr_value)
```

```
## 
## IQR: 68.5
```

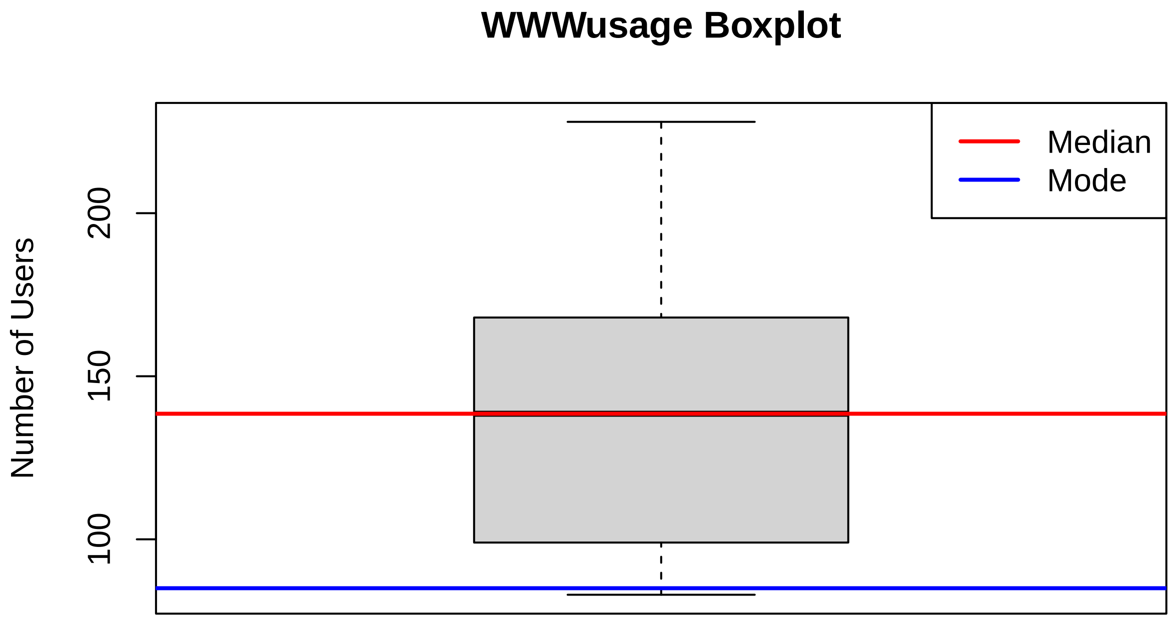
```r
cat("\nMode:", mode_value)
```

```
## 
## Mode: 85
```

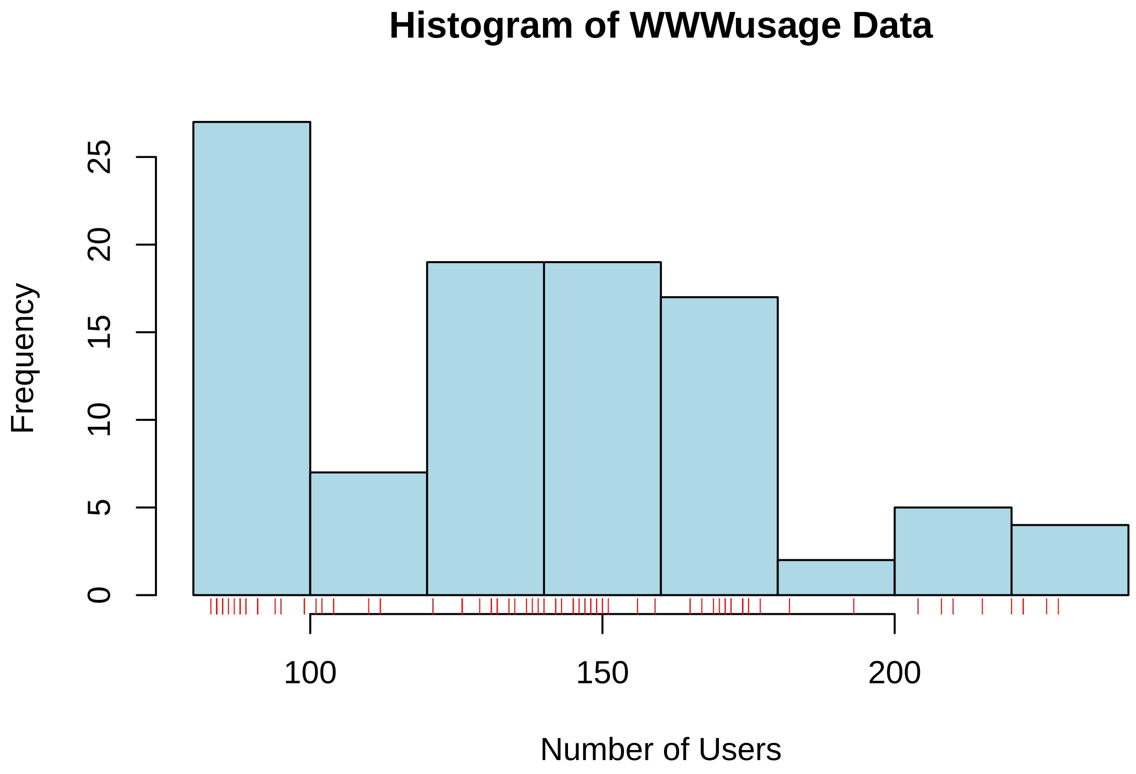
```
#(b)

# Calculate the lower and upper bounds for outliers
lower_bound <- q25 - 1.5 * iqr
upper_bound <- q75 + 1.5 * iqr

# Identify outliers
outliers <- WWWusage[WWWusage < lower_bound | WWWusage > upper_bound]

# Check outliers
if (length(outliers) > 0) {
  cat("There are", length(outliers), "outliers in the dataset.\n")
  cat("Outlier values:", outliers, "\n")
} else {
  cat("There are no outliers in the dataset.\n")
}
```

```
## There are no outliers in the dataset.
```

```
#(c)
# Boxplot
boxplot(WWWusage, main="WWWusage Boxplot", ylab="Number of Users")

# median and mode lines
abline(h = median_value, col = "red", lwd = 2)
abline(h = mode_value, col = "blue", lwd = 2)

# legend
legend("topright", legend=c("Median", "Mode"),
       col=c("red", "blue"), lwd=2)
```



```
#(d)
#histogram
histo <- hist(WWWusage, main = "Histogram of WWWusage Data",
              col = "lightblue", xlab = "Number of Users", ylab = "Frequency")

# Add a rug plot to visualize individual data points
rug(WWWusage, col = "red")
```



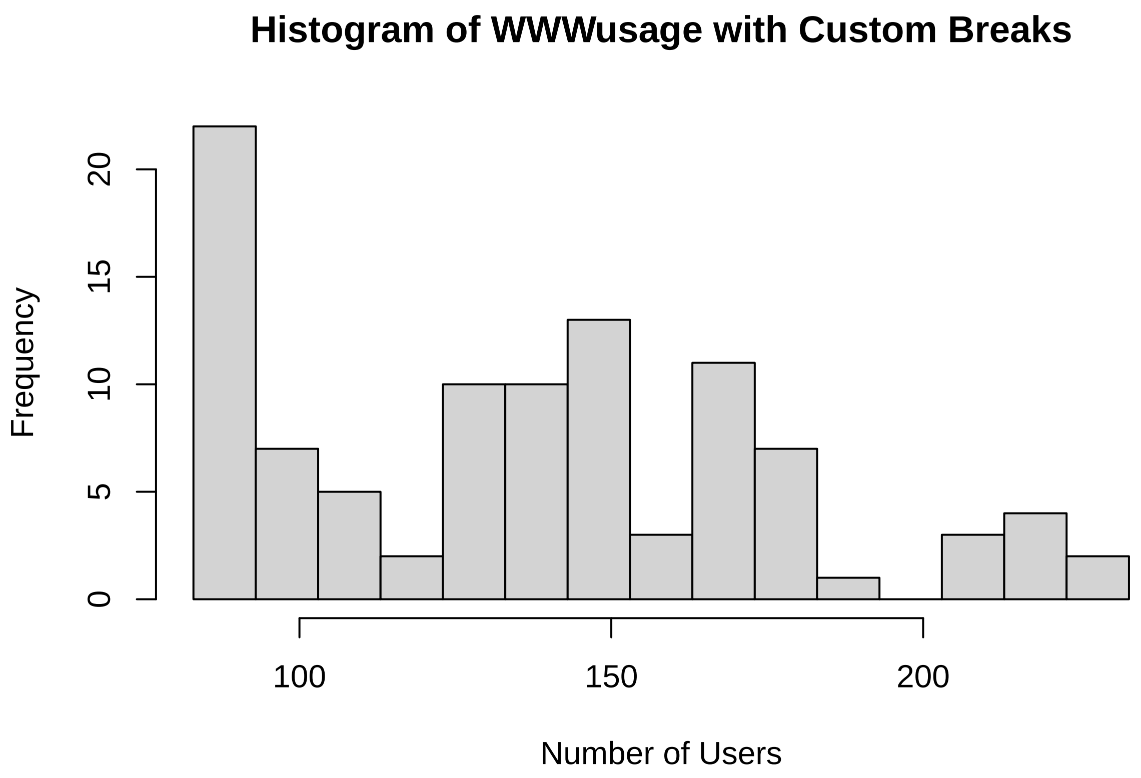
```r
# Get numerical information about the histogram
str(histo)
```

```
## List of 6
##  $ breaks  : int [1:9] 80 100 120 140 160 180 200 220 240
##  $ counts  : int [1:8] 27 7 19 19 17 2 5 4
##  $ density : num [1:8] 0.0135 0.0035 0.0095 0.0095 0.0085 0.001 0.0025 0.002
##  $ mids    : num [1:8] 90 110 130 150 170 190 210 230
##  $ xname   : chr "WWWusage"
##  $ equidist: logi TRUE
##  - attr(*, "class")= chr "histogram"
```

```r
# calculate bin size
bin_size <- diff(histo$breaks)
cat("Bin Size:", bin_size[1], "\n")
```

```
## Bin Size: 20
```

```r
#(e)
custom_breaks <- seq(min(WWWusage), max(WWWusage) + 10, by = 10)
#custom_breaks <- seq(min(WWWusage), max(WWWusage), by=10)
histo_custom <- hist(WWWusage, breaks=custom_breaks,
                     main="Histogram of WWWusage with Custom Breaks",
                     xlab="Number of Users")
```



```r
str(histo_custom)
```

```
## List of 6
##  $ breaks  : num [1:16] 83 93 103 113 123 133 143 153 163 173 ...
##  $ counts  : int [1:15] 22 7 5 2 10 10 13 3 11 7 ...
##  $ density : num [1:15] 0.022 0.007 0.005 0.002 0.01 0.01 0.013 0.003 0.011 0.007 ...
##  $ mids    : num [1:15] 88 98 108 118 128 138 148 158 168 178 ...
##  $ xname   : chr "WWWusage"
##  $ equidist: logi TRUE
##  - attr(*, "class")= chr "histogram"
```

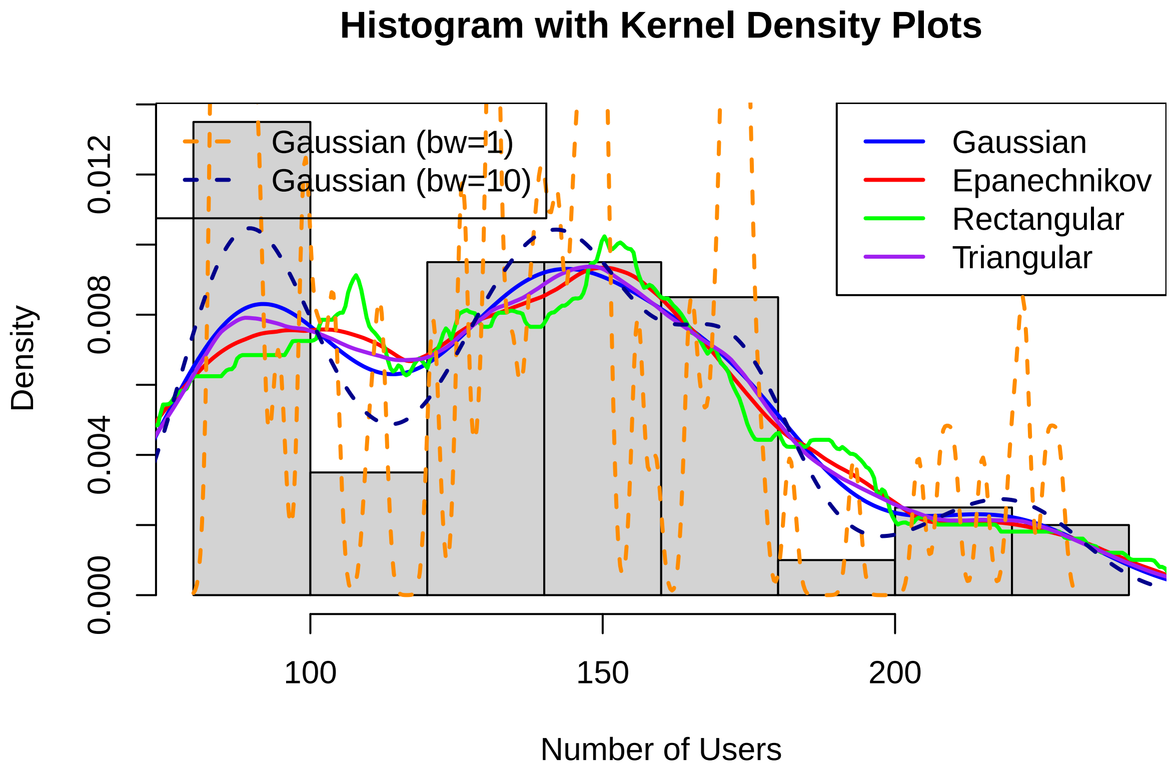
```r
bin_index <- which(histo_custom$breaks > 100 & histo_custom$breaks <= 110)
bin_count <- histo_custom$counts[bin_index - 1]

total_samples <- length(WWWusage)
probability_hist <- bin_count / total_samples

cat("Estimated probability (from histogram):", probability_hist, "\n")
```

```
## Estimated probability (from histogram): 0.07
```

```r
direct_count <- sum(WWWusage > 100 & WWWusage <= 110)
probability_direct <- direct_count / total_samples

cat("Actual probability (direct computation):", probability_direct, "\n")
```

```
## Actual probability (direct computation): 0.05
```

```r
#(f)
histo <- hist(WWWusage, probability = TRUE, main = "Histogram with Kernel Density Plots",
              xlab = "Number of Users", col = "lightgray")

# density plot with Gaussian kernel
density_gaussian <- density(WWWusage, kernel = "gaussian")
lines(density_gaussian, col = "blue", lwd = 2)

# density plot with Epanechnikov kernel
density_epanechnikov <- density(WWWusage, kernel = "epanechnikov")
lines(density_epanechnikov, col = "red", lwd = 2)

# density plot with Rectangular kernel
density_rectangular <- density(WWWusage, kernel = "rectangular")
lines(density_rectangular, col = "green", lwd = 2)

# density plot with Triangular kernel
density_triangular <- density(WWWusage, kernel = "triangular")
lines(density_triangular, col = "purple", lwd = 2)

# legend
legend("topright", legend = c("Gaussian", "Epanechnikov", "Rectangular", "Triangular"),
       col = c("blue", "red", "green", "purple"), lwd = 2)

# Varying Bandwidth

density_bw_small <- density(WWWusage, kernel = "gaussian", bw = 1)
lines(density_bw_small, col = "darkorange", lty = 2, lwd = 2)

density_bw_large <- density(WWWusage, kernel = "gaussian", bw = 10)
lines(density_bw_large, col = "darkblue", lty = 2, lwd = 2)

# legend for bandwidth variations
legend("topleft", legend = c("Gaussian (bw=1)", "Gaussian (bw=10)"),
       col = c("darkorange", "darkblue"), lty = 2, lwd = 2)
```


Histogram with Kernel Density Plots
Density
0.000
0.004
0.008
0.012
100
150
200
Number of Users
Gaussian (bw=1)
Gaussian (bw=10)
Gaussian
Epanechnikov
Rectangular
Triangular