SDA Group Submission Assignment Assign3

Group Gr18

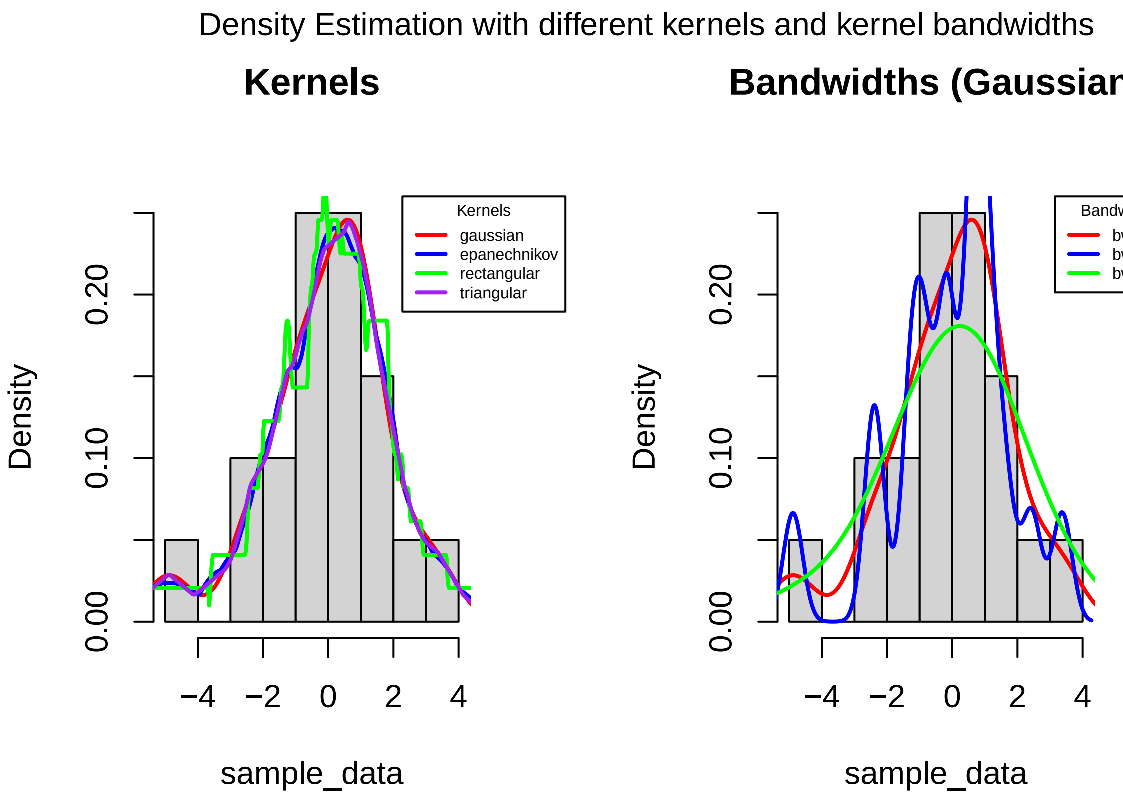
MengliFeng (2720589) and PepijnVanOostveen (2801582)

Exercise 1

a.

```r
# Set seed for reproducibility
set.seed(123)

# Generate random sample from t-distribution with 3 degrees of freedom
n <- 20
sample_data <- rt(n, df = 3)

# Define different kernel types and colors
kernels <- c("gaussian", "epanechnikov", "rectangular", "triangular")
colors_kernels <- c("red", "blue", "green", "purple")

# Define different bandwidth choices and colors
bandwidths <- c(density(sample_data)$bw, 0.3, 1.5)
colors_bandwidths <- c("red", "blue", "green")

# Adjust plot margins to make space for legends
par(mfrow = c(1, 2), mar = c(5, 4, 6, 4))  # Extra right margin for the legend

# Plot histogram with different kernel choices
hist(sample_data, probability = TRUE, main = "Kernels", col = "lightgray", border =
↪  "black")

for (i in seq_along(kernels)) {
  lines(density(sample_data, kernel = kernels[i]), col = colors_kernels[i], lwd = 2)
}

# Add legend outside the plot
legend("topright", inset = c(-0.3, 0), legend = kernels, col = colors_kernels, lwd = 2,
↪  cex = 0.5, title = "Kernels", xpd = TRUE)

# Plot histogram with different bandwidth choices
hist(sample_data, probability = TRUE, main = "Bandwidths (Gaussian)", col = "lightgray",
↪  border = "black")

for (i in seq_along(bandwidths)) {
  lines(density(sample_data, bw = bandwidths[i]), col = colors_bandwidths[i], lwd = 2)
}
```

```
# Add legend outside the plot
legend("topright", inset = c(-0.3, 0), legend = paste("bw =", round(bandwidths, 2)), col
↪  = colors_bandwidths, lwd = 2, cex = 0.5,
       title = "Bandwidths", xpd = TRUE)

# --- Add an Overall Title ---
mtext("Density Estimation with different kernels and kernel bandwidths",line = 4, cex =
↪  1, font = 1, adj = 1)
```



b.

From the generated plots, we can observe: • Effect of Kernel Choice: Different kernels produce similar overall shapes, but their smoothness varies slightly. The Gaussian kernel is the smoothest, while the rectangular kernel has more abrupt changes. • Effect of Bandwidth Choice: The bandwidth has a much larger influence than the kernel. A smaller bandwidth (0.3) captures more fluctuations in the data, while a larger bandwidth (1.5) smooths out more features. • Key Influence: Bandwidth choice has a bigger impact on the estimator compared to kernel choice.

c.

```
h_opt <- function(x) {
  sigma_hat <- min(sd(x), IQR(x) / 1.34)  # Compute standard deviation and interquartile
↪  range
  h_optimal <- 1.06 * sigma_hat * length(x)^(-1/5)  # Optimal bandwidth formula
  return(h_optimal)
}
```

```r
# Compute optimal bandwidth for the sample
h_opt_value <- h_opt(sample_data)

# Compare with R's default bandwidth
default_bw <- density(sample_data)$bw

# Print results
cat("Optimal Bandwidth (h_opt):", h_opt_value, "\n")
```

```
## Optimal Bandwidth (h_opt): 0.831087
```

```r
cat("Default R Bandwidth:", default_bw, "\n")
```

```
## Default R Bandwidth: 0.7056399
```

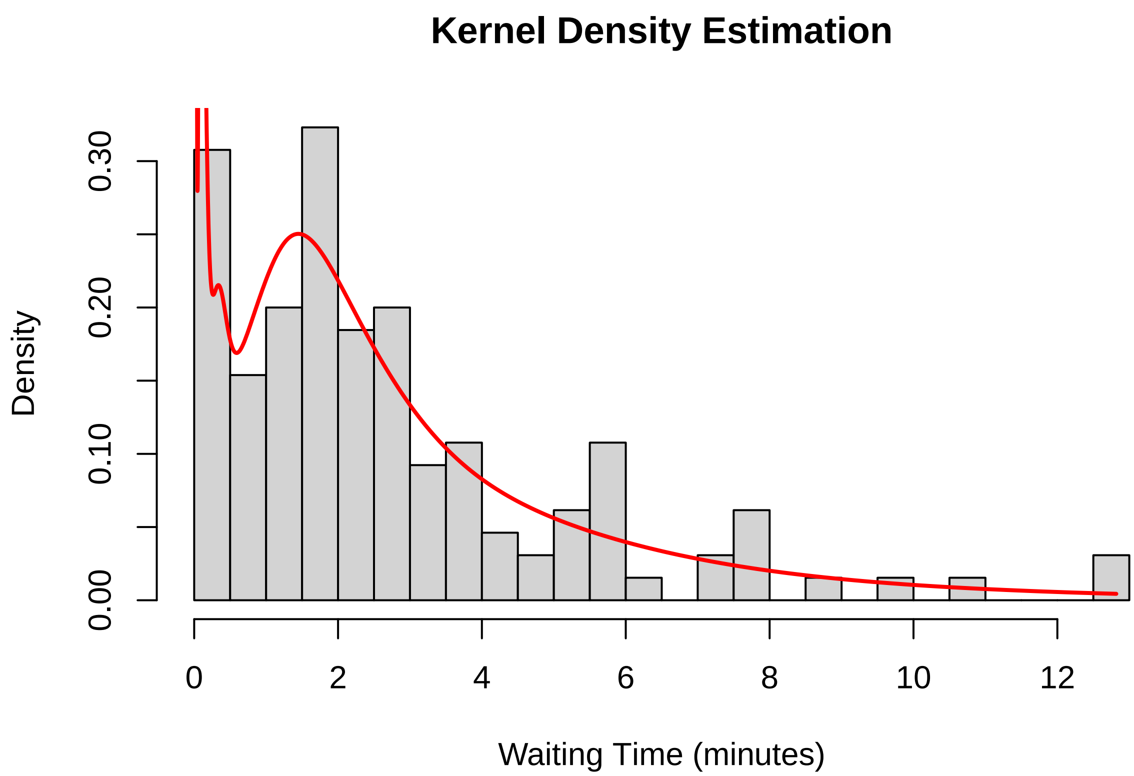
Exercise 2

a.

I expect that $f(t) = 0$ only for $t < 0$. As f is only 0 when there is no chance that you wait for that amount of time and you can't wait for a negative amount of time. For any other amount of time there is a non-zero chance as you can wait anywhere from 0 minutes (someone else called the elevator at the right time that you can enter the elevator with them) and a practically infinite amount of time (the elevator is out of order).

b.

```r
# load the dataset
wait_times <- readRDS("waiting_times.RDS")

# log-transform as suggested by the hint
log_times = log(wait_times)

# estimating the density of the transformed sample
y_seq <- seq(min(log_times), max(log_times), length.out = 512)
density_y <- density(log_times, from = min(y_seq), to = max(y_seq))

# Transform back to the original scale
x_seq <- exp(y_seq)
density_x <- density_y$y / exp(y_seq)

# plot histogram
hist(wait_times, probability = TRUE, breaks = 30, main = "Kernel Density Estimation",
↪  xlab = "Waiting Time (minutes)")

# Add KDE line
lines(x_seq, density_x, col = "red",lwd = 2)
```

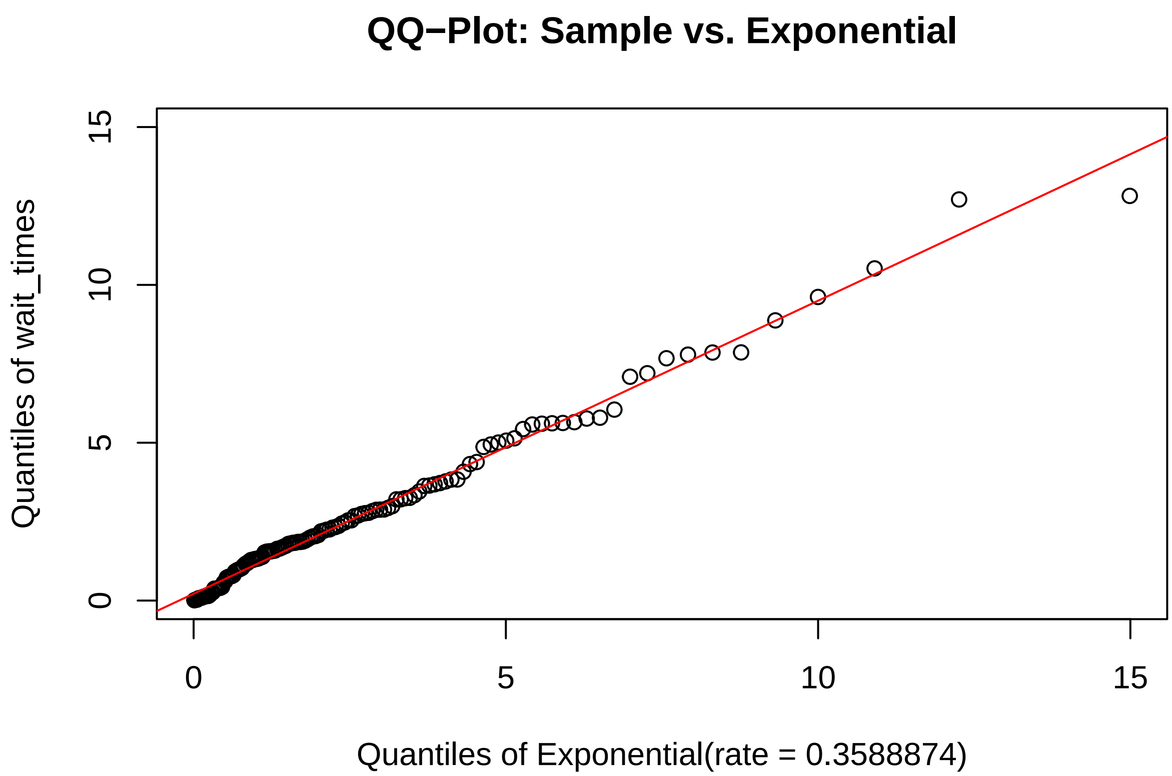


c.

```r
library(EnvStats)
```

```
##
## Attaching package: 'EnvStats'

## The following objects are masked from 'package:stats':
##
##     predict, predict.lm
```

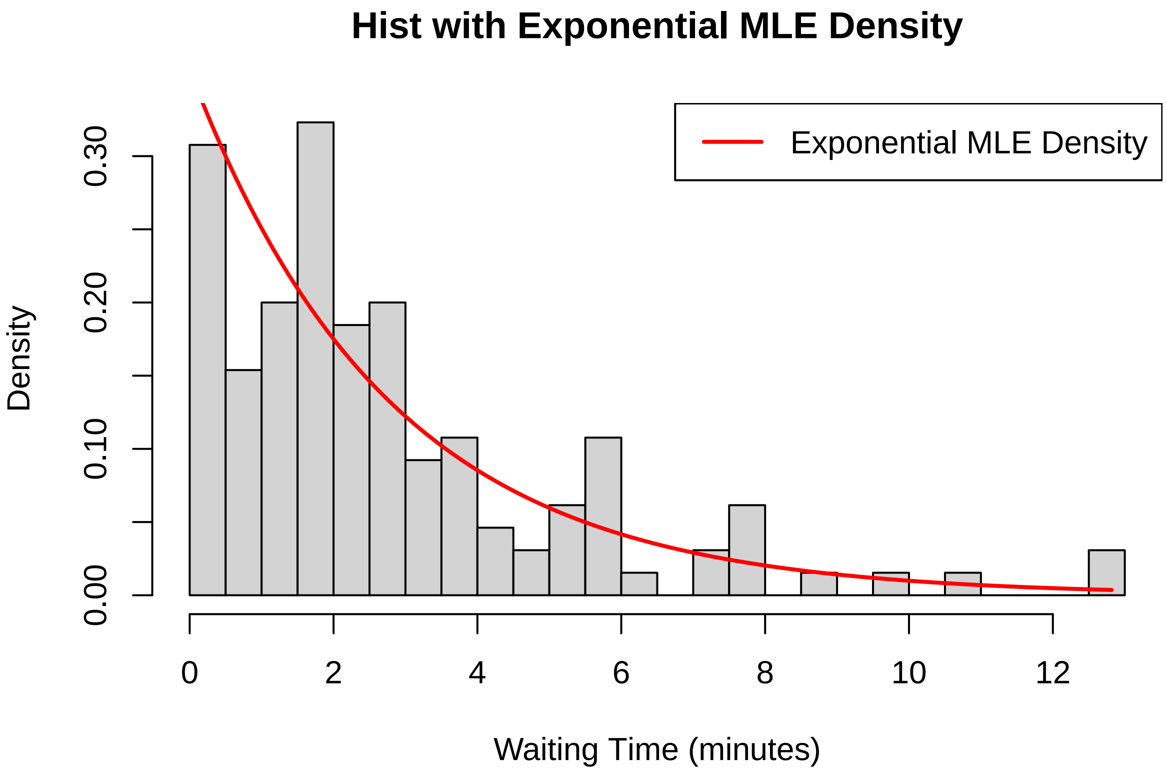
```r
# QQ-Plot for Exponential distribution
qqPlot(wait_times, distribution = "exp", estimate.params = TRUE, add.line = TRUE,
↪  line.col = "red", main = "QQ-Plot: Sample vs. Exponential")
```



The points seem to follow the line pretty well so I think that an exponential distribution is an appropriate choice to model the sample.

d.

```r
# We know that the rate MLE for the rate lambda is 1 divided by the sample mean (the
↪  waiting times)
l_mle <- 1 / mean(wait_times)

x_seq <- seq(min(wait_times), max(wait_times), length.out = 512)

# Now we make make the theoretical density function using our estimate
exp_density <- l_mle * exp(-l_mle * x_seq)

# Plot the histogram again
hist(wait_times, probability = TRUE, breaks = 30, main = "Hist with Exponential MLE
↪  Density", xlab = "Waiting Time (minutes)")

# Add a line for our theoretical density function
lines(x_seq, exp_density, col = "red", lwd = 2)
legend("topright", legend = c("Exponential MLE Density"), col = "red", lwd = 2)
```



The exponential density is worse than the estimator found b since the sample doesn't exactly follow an exponential distribution. Instead the sample follows an exponential distribution, but with with an added peak close to zero because relatively often you don't have to wait for the elevator at all because someone else has already called it.