Non-parametric Methods - Exam part II

Julien Colot

Project: Analyzing the 20 km Brussels Running Race Results Using Non-Parametric Methods
2022-2023

1st year Master of Statistics
Hasselt University

Submission Date: 11 August 2023

Lecturer:
Prof. Yudhie Andriyana

1 Introduction

The dataset for this analysis consists of the official results from the 2023 edition of the annual 20-km race in Brussels, held every May. My participation in that year's race sparked my interest in exploring this particular dataset. The data is publicly accessible on Chronorace, a platform for course results, but is only available in paginated HTML content. Consequently, the gathering of the data required downloading and extraction from HTML table to CSV format, page by page, through a script. The complete dataset is available in CSV and HTML tabular format at the links below:

GitHub repository: `https://github.com/jcolot/20km`

Chronorace: `https://prod.chronorace.be/result/sibp/Classement20km.aspx`

In 2023, the race attracted 41,314 participants, a testament to its popularity. The data reveals a diverse age and gender distribution, including 25,690 males and 15,568 females, along with 56 participants who preferred not to disclose their gender. Participant ages were collected through self-declaration, without the need for verification via identification documents. As a result, several participants who provided improbable ages outside the range of 5 to 95 years old were excluded from my dataset. By refining the data to include only complete cases—those containing age, disclosed gender, and course results—I was able to obtain a comprehensive dataset consisting of 36,517 entries (22,690 males and 13,827 females).

Participants' approach to the race widely varies. While some engage in competitive running, others choose a more leisurely pace. This decision may be influenced by various factors, such as personal preference, fatigue, or other considerations. Intriguingly, as we shall explore further, these preferences appear to be substantially shaped by both gender and age.

The aim of this analysis is to shed light on the race's dynamics through the application of non-parametric methods, specifically by estimating the influence of covariates gender and age on the course results.

2 Density Estimation

As a first step in the exploration of my dataset, I used histograms and kernel density estimation (KDE) of the covariates —gender and age— (figures 1 and 2) and to the response variable, the course result expressed in minutes. A gaussian kernel was used for the KDEs and the bandwidth was left to its default value from the `density()` function, which is 0.9 times the minimum of the standard deviation and the interquartile range divided by 1.34 times the sample size to the negative one-fifth power, also called Silverman's 'rule of thumb' since its introduction by Silverman (1986).

The data was organized into age groups, using ten-year intervals as midpoints, and separate Kernel Density Estimates (KDE) were fitted for each of these defined age brack-

ets (figure 3). This approach revealed specific trends in performance and preferences. A significant observation is the growing divergence between males and females in KDE estimates as age increases. This divergence seems to indicate a tendency for women to prefer a slower pace, while men tend to maintain a competitive attitude towards the race, even in the older age groups. This pattern highlights distinct gender behaviors in the context of racing, which may warrant further exploration.

An attempt was made to employ a 2D kernel density estimation using the `MASS::kde2d()` function (figures 4 and 5), providing a different perspective on the patterns previously observed in the KDEs segmented by age groups. This approach further emphasized the trends already noticed, reinforcing the findings of the analysis.

3 Qantile Regression

In order to observe the effects of age on the best competitive runners, several quantile regressions where fitted, for percentile 5, 25, 50, 75 and 95 using qualitatively constrained quantile smoothing splines with the COBS library by Ng and Maechler (2007). Interestingly, the performance of the best runners seems relatively stable on the range 20 to 50 years old, with an optimum for runners in their late twenties (figure 6 and figure 7). The results are affected by the problem of crossingness in some part of the age range for females, due to the scarcity of older participants.

4 Nadaraya-Watson Estimator

The dataset was also analyzed using kernel regression, applying the estimator initially found separately by Nadaraya (1964) and Watson (1964), to determine the expected course results as a function of age. This analysis was performed using the `ksmooth` function in R, utilizing a Gaussian kernel. The resulting graph reveals a peak in course results for participants in their early twenties. Following this peak, the expected performance gradually diminishes for both males and females. As the analysis nears the older participants in the dataset, the curves begin to exhibit more fluctuations, likely a consequence of limited data within those age ranges. It should be noted that these plots do not capture the high variability of the data that could be seen in the 2D histograms.

5 Conclusion

This analysis of race results has revealed specific trends among genders and age groups. Women, compared to men, tend to run less competitively and engage in the 20 km race less often, with a notable shift towards a slower pace as they age. This trend is less pronounced in men. The findings may suggest differences in aging between genders, or changes in

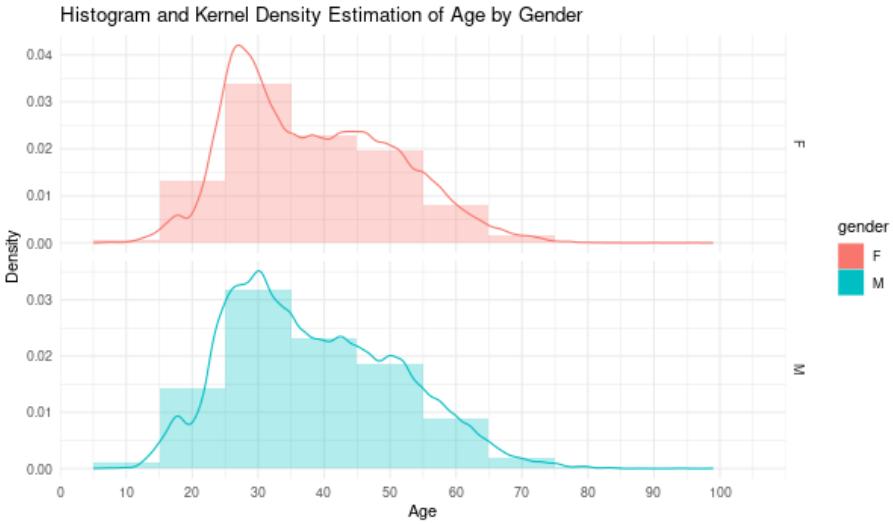

Figure 1: Age proportion histogram and kernel density estimation

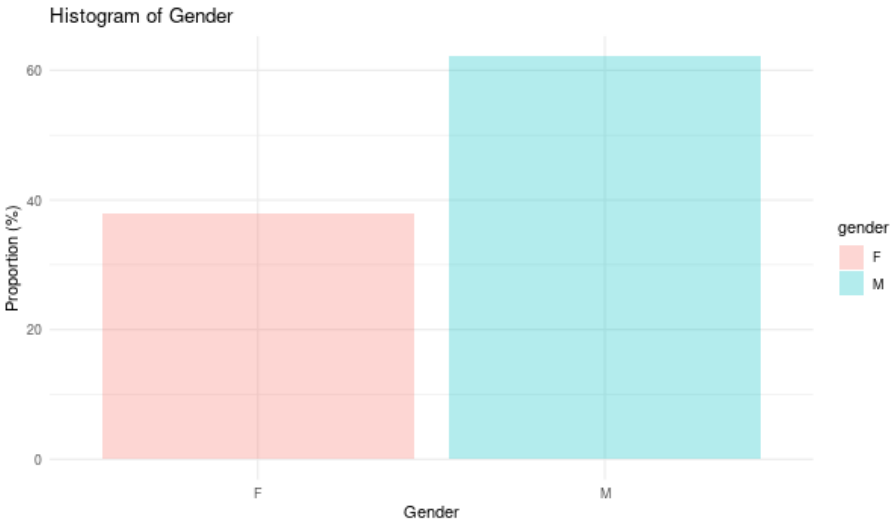

Figure 2: Gender proportion histogram

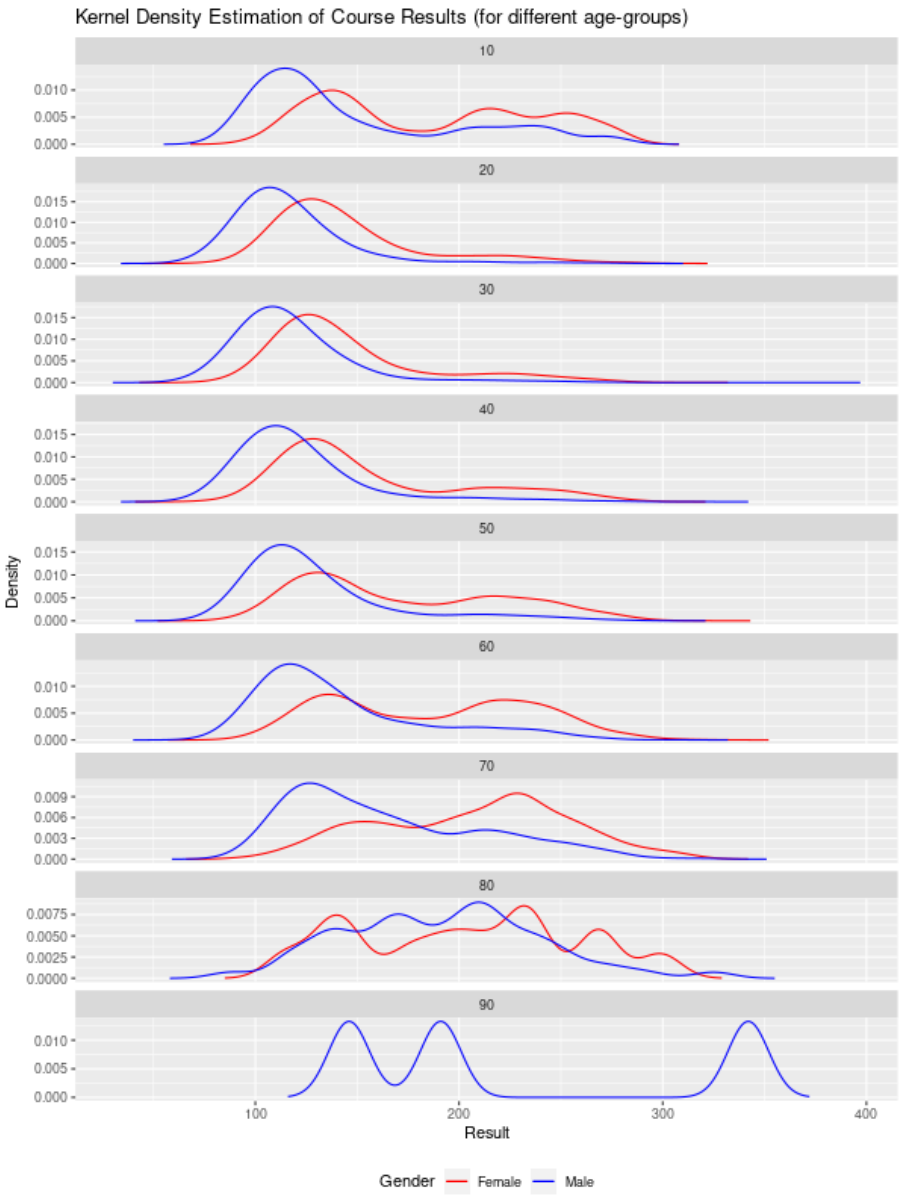

Figure 3: Kernel density estimation by age-group

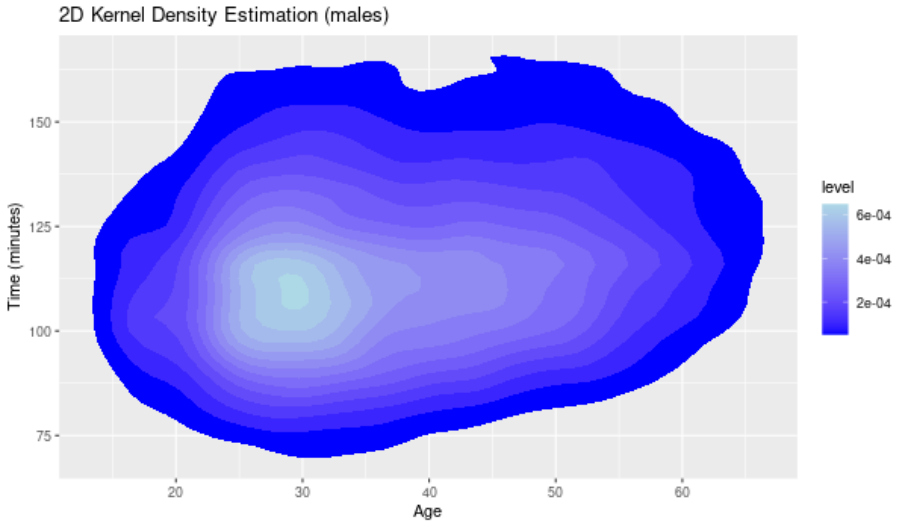

Figure 4: Two-dimensional kernel density estimation with `kde2` (males)

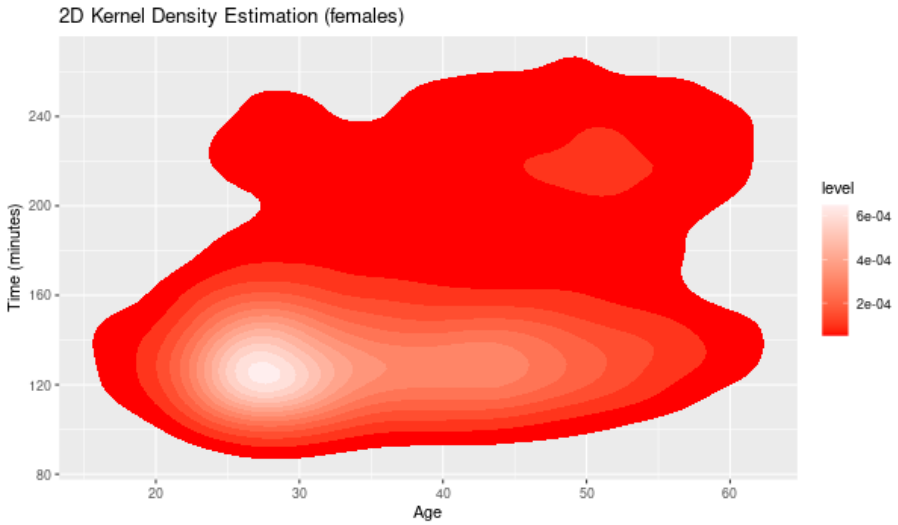

Figure 5: Two-dimensional kernel density estimation with `kde2` (females)

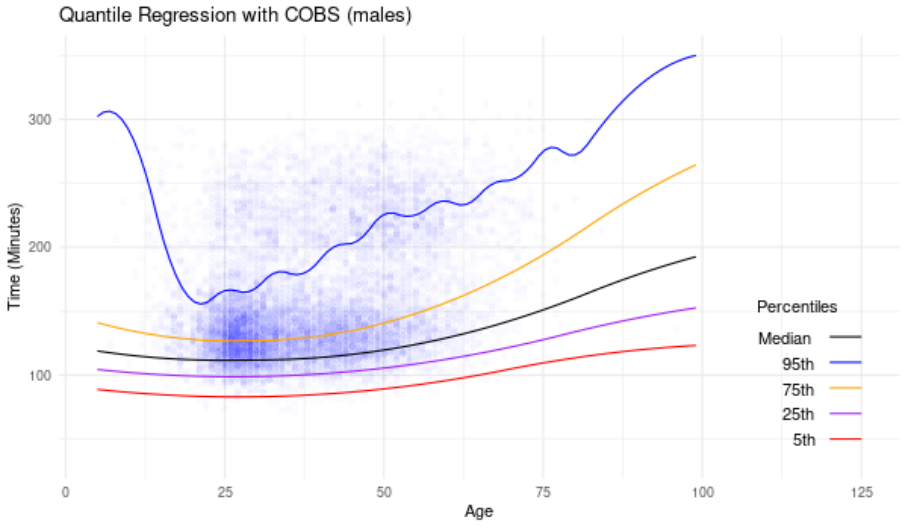

Figure 6: Quantile regression with `COBS` (males)

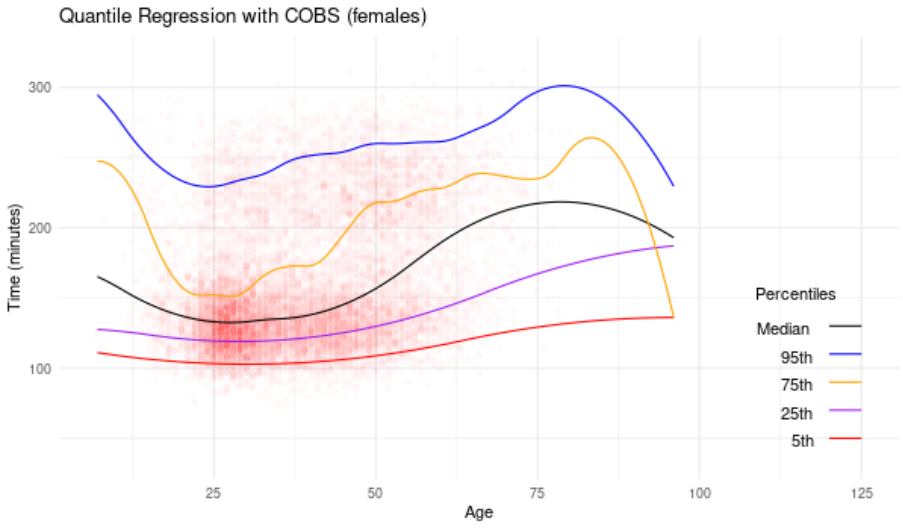

Figure 7: Quantile regression with `COBS` (females)

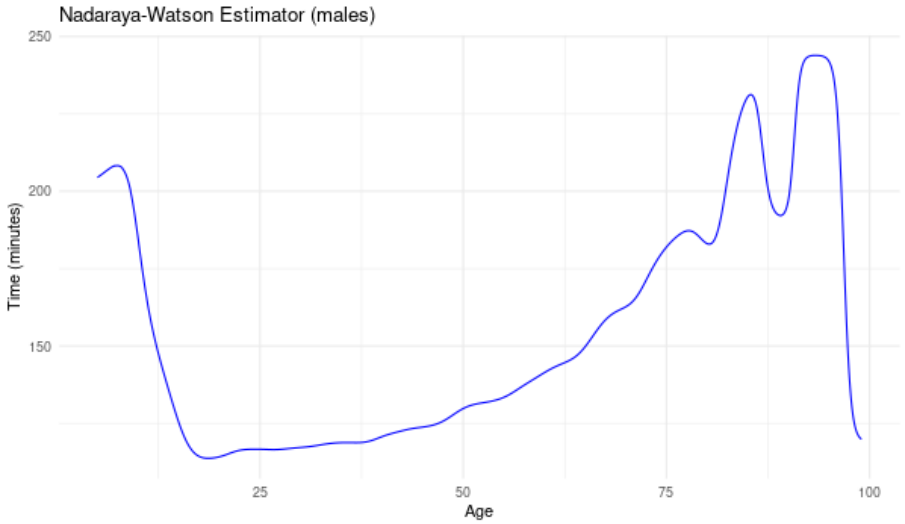

Figure 8: Kernel regression using the Nadaraya-Watson estimator

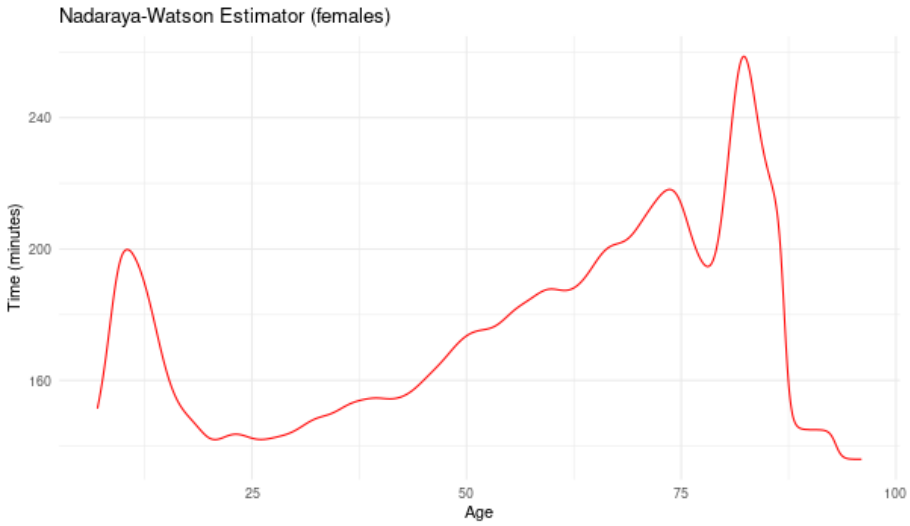

Figure 9: Kernel regression using the Nadaraya-Watson estimator

preference influenced by sociocultural factors. Additionally, runner performance seems to peak in the late twenties and then slowly decline. These insights could have practical implications in understanding gender dynamics in competitive and leisure sport activities.

References

Elizbar A Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9 (1):141–142, 1964.

Pin Ng and Martin Maechler. A fast and efficient implementation of qualitatively constrained quantile smoothing splines. *Statistical Modelling*, 7(4):315–328, 2007.

Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

Geoffrey S Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 359–372, 1964.

Appendix - R code

````
---
title: "20km"
author: "Julien Colot"
date: "2023-08-04"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r read_csv}
library(ggplot2)
library(readr)
library(tidyverse)
library(QRegVCM)
library(cobs)


# Load the data from the CSV file
df <- read_csv('20km.csv');
df_with_uncomplete_cases <- df;
gender_count <- table(df$gender);

df$age <- as.numeric(df$x_age);
df <- df[!is.na(df$age), ]
df <- df[!is.na(df$time_seconds), ]

# Summary of complete case participants by gender

df_age_above_100 <- df[df$age > 100,];
df_age_under_5 <- df[df$age < 5,];

df <- df[df$age >= 5 & df$age <= 100 & df$gender != 'X', ]
gender_count_complete <- table(df$gender);
print(gender_count_complete);

# Create age group
````
````

```
df$age_group <- cut(df$age, breaks = seq(5, 105, 10), include.lowest = TRUE)
df$age_group_midpoint <- sapply(str_extract_all(df$age_group, "\\d+"),

                                function(x) mean(as.numeric(x)))

df$time_seconds <- as.numeric(df$time_seconds)

# Convert time_seconds into minutes and round
df$time_minutes <- round(df$time_seconds / 60)

df_initial <- df;

df_male <- df %>% filter(gender=='M');
df_male_initial <- df_male;

df_female <- df %>% filter(gender=='F');
df_female_initial <- df_female;


# Summarize the data by age_group
age_summary <- df %>%
  group_by(age_group_midpoint, gender) %>%
  summarize(count = n())

p <- ggplot(df_initial, aes(x = age)) +
  geom_histogram(aes(y = ..density.., fill = gender), position = "dodge",
                 binwidth = 10, alpha = 0.3) +
  geom_density(aes(color = gender), adjust = 0.5) +  # KDE line
  labs(title = "Histogram and Kernel Density Estimation of Age by Gender",
       x = "Age",
       y = "Density") +
  scale_x_continuous(breaks = seq(0, 100, 10)) +
  theme_minimal() +
  facet_grid(gender ~ ., scales = "free_y")

png(file="~/UHasselt/Non\ Parametric/hist_kde_age.png",
width=600, height=350)
# Print the plot
print(p)
dev.off();
```

```
df_gender_proportion <- df_initial %>%
  group_by(gender) %>%
  summarize(proportion = n() / nrow(df_initial) * 100)

# Create the plot
p <- ggplot(data = df_gender_proportion,
            aes(x = gender, y = proportion, fill = gender)) +
  geom_bar(stat = "identity", position = "dodge", alpha = 0.3) +
  labs(title = "Histogram of Gender",
       x = "Gender",
       y = "Proportion (%)") +
  scale_x_discrete() +
  theme_minimal()

png(file="~/UHasselt/Non\ Parametric/hist_gender.png",
width=600, height=350)
# Print the plot
print(p)
dev.off();
```

```{r}

# Select only necessary columns
df <- df %>%
  select(age_group_midpoint, time_minutes)

age_list <- split(df$time_minutes, df$age_group_midpoint)


# For df_male
# Select only necessary columns
df_male <- df_male %>%
  select(age_group_midpoint, time_minutes)

age_list_male <- split(df_male$time_minutes, df_male$age_group_midpoint)

density_list_male <- lapply(age_list_male, function(x) {
```

```
  x <- x[complete.cases(x)]
  if(length(x) > 1)
    density(x, bw = 10)
  else
    NULL
})

# Convert density_list into a data frame
df_density_male <- do.call(rbind, lapply(1:length(density_list_male), function(i) {
  if (!is.null(density_list_male[[i]])) {
    data.frame(
      age = rep(as.numeric(names(density_list_male)[i]),
                length(density_list_male[[i]]$x)), # use actual midpoint value
      result = density_list_male[[i]]$x,
      density = density_list_male[[i]]$y
    )
  }
}))

# For df_female
# Select only necessary columns
df_female <- df_female %>%
  select(age_group_midpoint, time_minutes)

age_list_female <- split(df_female$time_minutes, df_female$age_group_midpoint)

density_list_female <- lapply(age_list_female, function(x) {
  x <- x[complete.cases(x)]
  if(length(x) > 1)
    density(x, bw = 10)
  else
    NULL
})

# Convert density_list into a data frame
df_density_female <- do.call(rbind, lapply(1:length(density_list_female),
function(i) {
  if (!is.null(density_list_female[[i]])) {
    data.frame(
      age = rep(as.numeric(names(density_list_female)[i]),
```

```
                length(density_list_female[[i]]$x)), # use actual midpoint value
      result = density_list_female[[i]]$x,
      density = density_list_female[[i]]$y
    )
  }
}))

```

```{r, fig.height=12, fig.width=8}

df_density_male$gender <- "Male"
df_density_female$gender <- "Female"
df_density_combined <- rbind(df_density_male, df_density_female)

p <- ggplot() +
  geom_line(data = subset(df_density_combined, gender == "Female"),
            aes(x = result, y = density, color = "Female")) +
  geom_line(data = subset(df_density_combined, gender == "Male"),
            aes(x = result, y = density, color = "Male")) +
  facet_wrap(~age, ncol = 1, scales = "free_y") +
  labels(title = "Kernel Density Estimation of Course Results (for different age-groups)", x =
  scale_color_manual(values = c("Female" = "red", "Male" = "blue"),
                     name = "Gender") +
  theme(legend.position = "bottom")

png(file="~/UHasselt/Non\ Parametric/density-time_by_age_group.png",
width=600, height=800)
# Print the plot
print(p)
dev.off();


```

```{r}
p <- ggplot(df_male_initial, aes(x=age, y=time_minutes))+
    stat_density2d(aes(fill=..level..), geom="polygon") +
    scale_fill_gradient(low="blue", high="lightblue") +
    labels(title="2D Kernel Density Estimation (males)",
```

```
          x="Age",
          y="Time (minutes)");

png(file="~/UHasselt/Non\ Parametric/2d_density_males.png",
width=600, height=350)
# Print the plot
print(p)
dev.off();
```

```{r}
p <- ggplot(df_female_initial, aes(x=age, y=time_minutes))+
    stat_density2d(aes(fill=..level..), geom="polygon") +
    scale_fill_gradient(low = "#FF0000", high = "#FFEEEE") +
    labs(title="2D Kernel Density Estimation (females)",
          x="Age",
          y="Time (minutes)");


png(file="~/UHasselt/Non\ Parametric/2d_density_females.png",
width=600, height=350)
# Print the plot
print(p)
dev.off();

```

```{r}
library(cobs)

# reset to the the initial dataset, not grouped by age-groups
df_male <- df_male_initial;

x <- df_male$age
y <- df_male$time_minutes

# Specify pointwise constraints (boundary conditions) - adjust as needed
con <- rbind(c(-1, min(x), 350),
             c(-1, max(x), 350),
             c(1, min(x), 0))
```

```
# Compute and save models for different quantiles
quantiles <- c(0.05, 0.25, 0.5, 0.75, 0.95) # Common quantile values
for(tau in quantiles) {
  model_name <- paste0("Sbs_", tau * 100, "_male")
  pred_name <- paste0("df_", model_name, "_predictions")
  model = cobs(x, y, constraint="none", pointwise=con, lambda=-1, tau=tau)
  assign(model_name, model)
  predictions <- predict(model)
  predictions <- as.data.frame(cbind(predictions[,1], predictions[,2]))
  colnames(predictions) <- c("age", "time_minutes")
  assign(pred_name, predictions)
}

```

```{r}
#### Same for females

# reset to the the initial dataset, not grouped by age-groups
df_female <- df_female_initial;
x <- df_female$age
y <- df_female$time_minutes

# Specify pointwise constraints (boundary conditions) - adjust as needed
con <- rbind(c(-1, min(x), 350),
             c(-1, max(x), 350),
             c(1, min(x), 0))


# Compute and save models for different quantiles
quantiles <- c(0.05, 0.25, 0.5, 0.75, 0.95) # Common quantile values
for(tau in quantiles) {
  model_name <- paste0("Sbs_", tau * 100, "_female")
  pred_name <- paste0("df_", model_name, "_predictions")
  model = cobs(x, y, constraint="none", pointwise=con, lambda=-1, tau=tau)
  assign(model_name, model)
  predictions <- predict(model)
  predictions <- as.data.frame(cbind(predictions[,1], predictions[,2]))
```

```
  colnames(predictions) <- c("age", "time_minutes")
  assign(pred_name, predictions)
}
```

```{r}

print(p)

# Create the plot with ggplot2
p <- ggplot(df_female) +
  # Scatterplot of original data
     geom_point(aes(y=time_minutes, x=age), alpha=0.01, color="blue") +
  # Lines for median
     geom_line(data=df_Sbs_50_male_predictions, aes(x=age, y=time_minutes), color="black") +
  # Lines for 5th percentile
     geom_line(data=df_Sbs_5_male_predictions, aes(x=age, y=time_minutes), color="red") +
  # Lines for 95th percentile
     geom_line(data=df_Sbs_95_male_predictions, aes(x=age, y=time_minutes), color="blue") +
  # Lines for 25th percentile
     geom_line(data=df_Sbs_25_male_predictions, aes(x=age, y=time_minutes), color="purple") +
   # Lines for 75th percentile
     geom_line(data=df_Sbs_75_male_predictions, aes(x=age, y=time_minutes), color="orange") +
     annotate("text", x=115, y=140, label="Percentiles", vjust=-1) +
     annotate("segment", x=120, xend=125, y=130, yend=130, colour="black") +
     annotate("text", x=113, y=115, label="Median", vjust=-1) +
     annotate("segment", x=120, xend=125, y=110, yend=110, colour="blue") +
     annotate("text", x=115, y=95, label="95th", vjust=-1) +
     annotate("segment", x=120, xend=125, y=90, yend=90, colour="orange") +
     annotate("text", x=115, y=75, label="75th", vjust=-1) +
     annotate("segment", x=120, xend=125, y=70, yend=70, colour="purple") +
     annotate("text", x=115, y=55, label="25th", vjust=-1) +
     annotate("segment", x=120, xend=125, y=50, yend=50, colour="red") +
     annotate("text", x=116, y=35, label="5th", vjust=-1) +
     labs(color="Percentiles",title="Quantile Regression with COBS (males)",
          x="Age",
          y="Time (Minutes)") +
     theme_minimal();

png(file="~/UHasselt/Non\ Parametric/quantile_regression_males.png",
```

```
width=600, height=350)
# Print the plot
print(p)
dev.off();

# Create the plot with ggplot2
p <- ggplot(df_female) +
   # Scatterplot of original data
     geom_point(aes(y=time_minutes, x=age), alpha=0.01, color="red") +
   # Lines for median
     geom_line(data=df_Sbs_50_female_predictions, aes(x=age, y=time_minutes), color="black"
  # Lines for 5th percentile
     geom_line(data=df_Sbs_5_female_predictions, aes(x=age, y=time_minutes), color="red") +
  # Lines for 95th percentile
     geom_line(data=df_Sbs_95_female_predictions, aes(x=age, y=time_minutes), color="blue")
  # Lines for 25th percentile
     geom_line(data=df_Sbs_25_female_predictions, aes(x=age, y=time_minutes), color="purple
   # Lines for 75th percentile
     geom_line(data=df_Sbs_75_female_predictions, aes(x=age, y=time_minutes), color="orange
     annotate("text", x=115, y=140, label="Percentiles", vjust=-1) +
     annotate("segment", x=120, xend=125, y=130, yend=130, colour="black") +
     annotate("text", x=113, y=115, label="Median", vjust=-1) +
     annotate("segment", x=120, xend=125, y=110, yend=110, colour="blue") +
     annotate("text", x=115, y=95, label="95th", vjust=-1) +
     annotate("segment", x=120, xend=125, y=90, yend=90, colour="orange") +
     annotate("text", x=115, y=75, label="75th", vjust=-1) +
     annotate("segment", x=120, xend=125, y=70, yend=70, colour="purple") +
     annotate("text", x=115, y=55, label="25th", vjust=-1) +
     annotate("segment", x=120, xend=125, y=50, yend=50, colour="red") +
     annotate("text", x=116, y=35, label="5th", vjust=-1) +
     labs(color="Percentiles",title="Quantile Regression with COBS (females)",
          x="Age",
          y="Time (minutes)") +
     theme_minimal();

png(file="~/UHasselt/Non\ Parametric/quantile_regression_females.png",
width=600, height=350)
# Print the plot
print(p)
dev.off();
```

```
```

```{r}

### Nadaraya-Watson Kernel Regression

ksmooth_fit_male <- as.data.frame(ksmooth(df_male$age, df_male$time_minutes, "normal",
                                          bandwidth = 4));
colnames(ksmooth_fit_male) <- c('age','time');

p <- ggplot() +
  geom_line(data = ksmooth_fit_male,
            aes(x = age, y = time), color = "blue") +
  labs(title = "Nadaraya-Watson Estimator (males)",
       x = "Age", y = "Time (minutes)") +
  theme_minimal();

png(file="~/UHasselt/Non\ Parametric/ksmooth_males.png",
width=600, height=350)

print(p)
dev.off()

ksmooth_fit_female <- as.data.frame(ksmooth(df_female$age, df_female$time_minutes,
                                            "normal", bandwidth = 4));
colnames(ksmooth_fit_female) <- c('age','time');


p <- ggplot() +
  geom_line(data = ksmooth_fit_female,
            aes(x = age, y = time), color = "red") +
  labs(title = "Nadaraya-Watson Estimator (females)",
       x = "Age", y = "Time (minutes)") +

  theme_minimal();

png(file="~/UHasselt/Non\ Parametric/ksmooth_females.png",
width=600, height=350)
print(p)
dev.off()
```

```
```