Practical Question - PS2

Pedro Henrique

26/10/2021

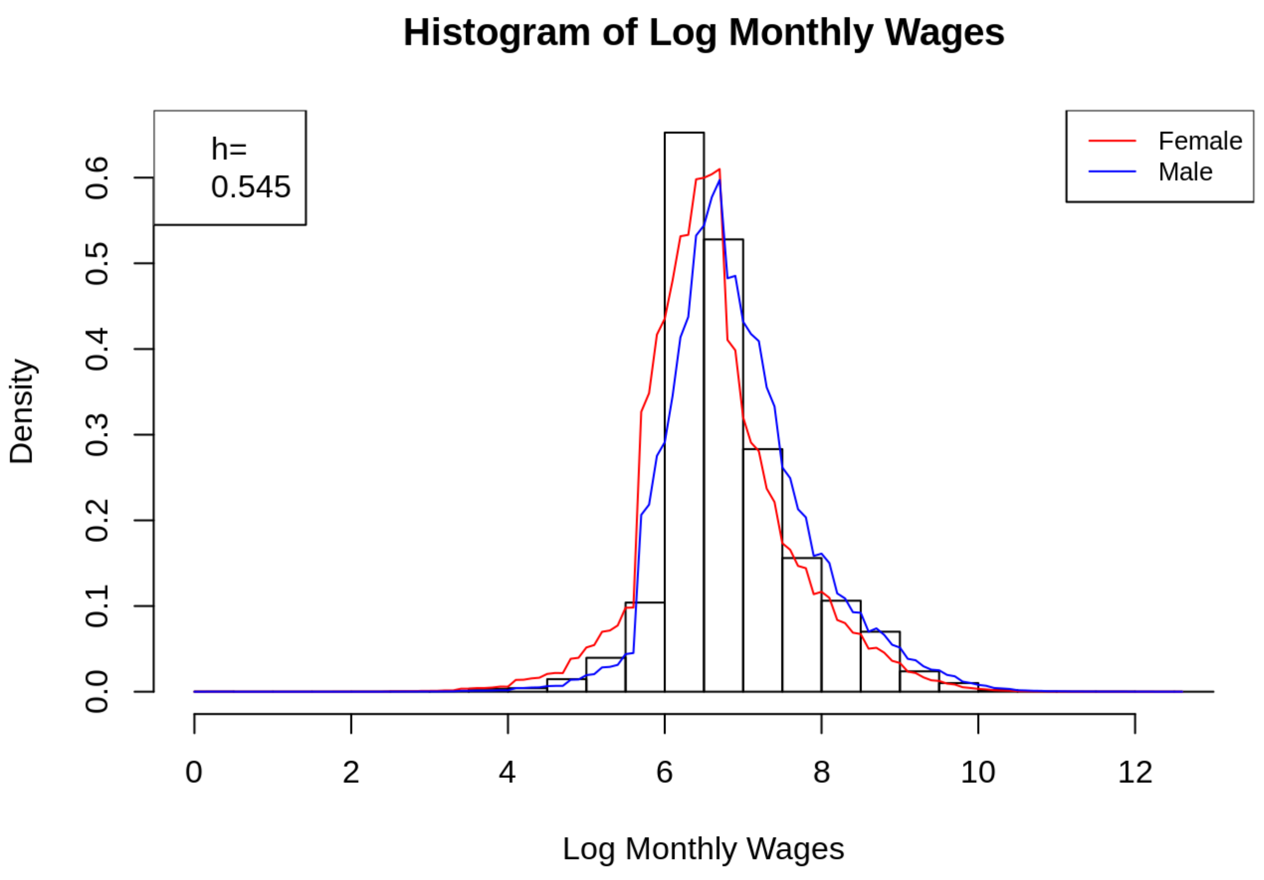
Question 6

Let’s use a uniform kernel to build a kernel density estimator function:

```r
# Uniform kernel function
uniform_kernel_function <- function(z){
  K <- 0.5 * as.numeric(abs(z)<1)
  return(K)
}
```

Given the sample code showed in class, for this question, it’s better not to reinvent the wheel. Therefore, we adapt the sample code. We end up with:

```r
# Kernel density estimator for each sex
kernel_density_estimator_male <- function(x){
  z <- (logearn_m - x)/h
  K <- uniform_kernel_function(z)
  s <- sum(K)
  f_hat <- s/(Nm*h)
  return(f_hat)
}

kernel_density_estimator_female <- function(x){
  z <- (logearn_f - x)/h
  K <- uniform_kernel_function(z)
  s <- sum(K)
  f_hat <- s/(Nf*h)
  return(f_hat)
}

# We can construct a vector of values to catch various data points of our variable
x <- seq(min(df$log_monthly_wages), max(df$log_monthly_wages),0.1)

# Log monthly earnings for males and sample size
logearn_m <- df %>% filter(female == 0) %>% select(log_monthly_wages) %>% pull()
Nm <- length(logearn_m)

# Log monthly earnings for females and sample size
logearn_f <- df %>% filter(female == 1) %>% select(log_monthly_wages) %>% pull()
Nf <- length(logearn_f)

# Estimated kernel values
h <- 0.545
yf <- sapply(x, kernel_density_estimator_female)
ym <- sapply(x, kernel_density_estimator_male)

# The plot
`Log Monthly Wages` <- df$log_monthly_wages
hist(`Log Monthly Wages`, freq=FALSE)
lines(x, yf, col = "red")
lines(x, ym, col = "blue")
legend("topright", legend=c("Female", "Male"), col=c("red", "blue"), lty=1, cex=0.
8)
legend("topleft", legend=c("h=", h))
```



Question 7

The optimum bandwidth is given by:

$$h^* = \delta\left(\int f''(x_0)^2\,dx_0\right)^{0.2} N^{-0.2}$$

Assuming that our data is normally distributed, then:

$$\int f''(x_0)^2\,dx_0 = \frac{3}{8\sqrt{\pi}\sigma^5} = \frac{0.2116}{\sigma^5}$$

Therefore, the optimal bandwidth is:

$$h^* = \delta\left(\frac{0.2116}{\sigma^5}\right)^{0.2} N^{-0.2} = 1.3643\delta N^{-0.2} s$$

The sample standard deviation for the general data and for each sex are:

```
s <- sd(df$log_monthly_wages)
s
```

```
## [1] 0.85
```

```
s_m <- sd(logearn_m)
s_m
```

```
## [1] 0.835
```

```
s_f <- sd(logearn_f)
s_f
```

```
## [1] 0.841
```

From table 9.1 of Cameron et Privadi (2005), we have the value of δ. Hence, the value of the optimal bandwidths for male and female, respectivaly, are:

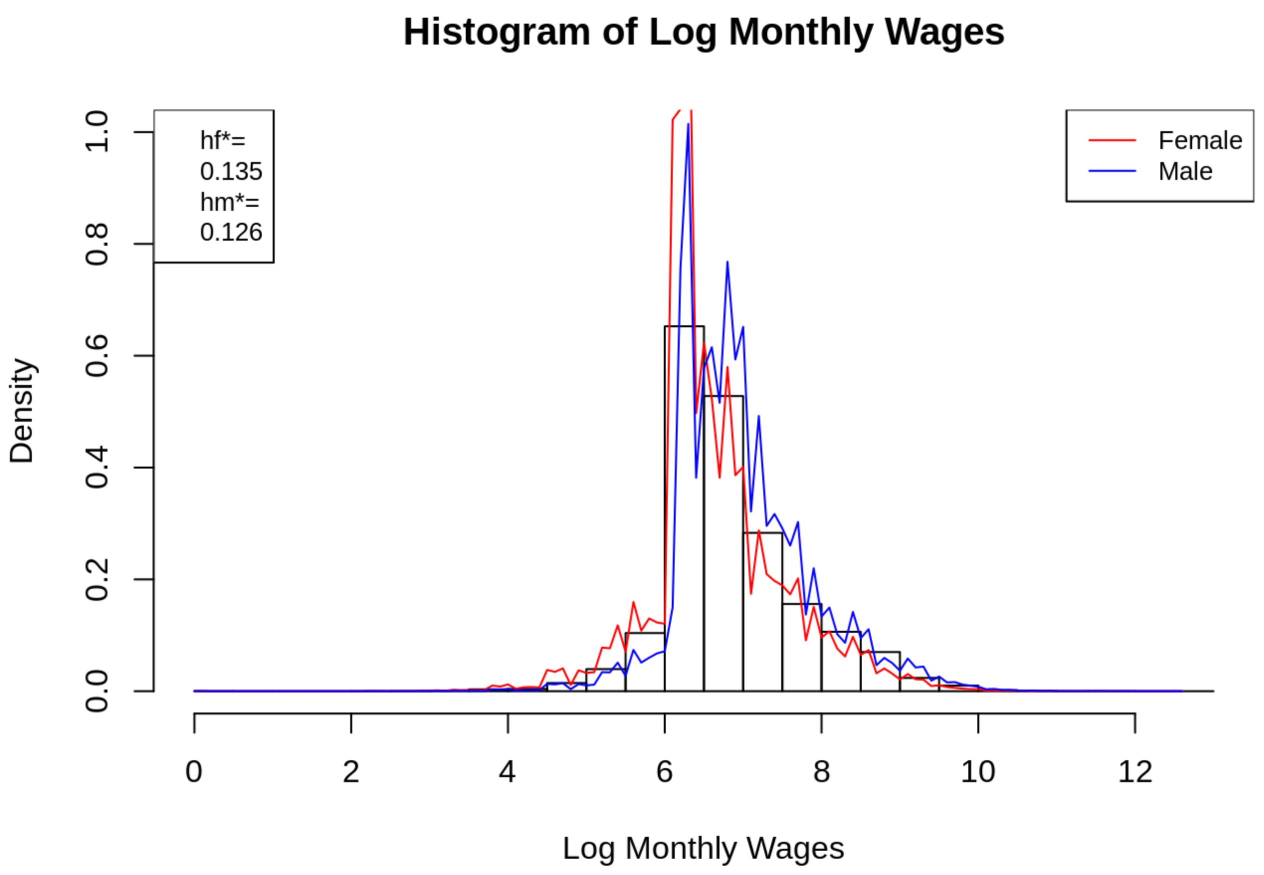
```
h_star_m <- 1.3643*1.3510*(Nm^-0.2)*s_m
h_star_m
```

```
## [1] 0.126
```

```
h_star_f <- 1.3643*1.3510*(Nf^-0.2)*s_f
h_star_f
```

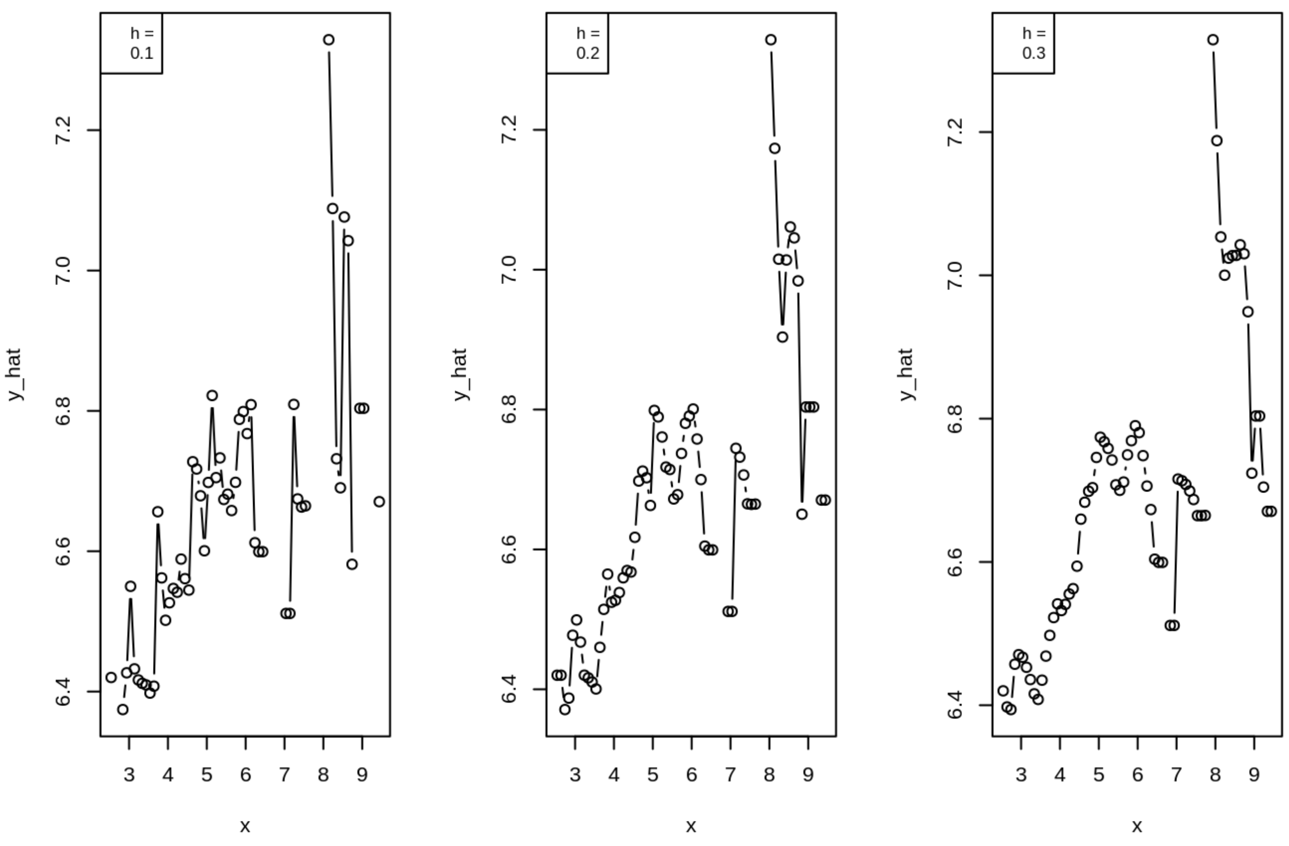
```
## [1] 0.135
```

Recalculating the estimated results, we end up with:



Question 8

```r
# Epanechnikov kernel
epanechnikov <- function(z) {0.75*(1-z^2)*as.numeric(abs(z)<1)}

# Our data
log_density <- df$log_density
log_monthly_wages <- df$log_monthly_wages

# Local Constant Estimator or Kernel regression
local_constant_estimator <- function(x) {
  z = (log_density - x)/h
  k <- epanechnikov(z)
  m_hat <- sum(k*log_monthly_wages)/sum(k)
  return(m_hat)
}

x <- seq(min(log_density), max(log_density), 0.1)

# Plots
par(mfrow = c(1,3))
# Bandwith = .1
h <- 0.1
y_hat <- sapply(x, local_constant_estimator)
plot(x, y_hat, type = 'b')
legend("topleft", legend=c("h =", h), cex=0.8)

# Bandwith = .2
h <- 0.2
y_hat <- sapply(x, local_constant_estimator)
plot(x, y_hat, type = 'b')
legend("topleft", legend=c("h =", h), cex=0.8)

# Bandwith = .3
h <- 0.3
y_hat <- sapply(x, local_constant_estimator)
plot(x, y_hat, type = 'b')
legend("topleft", legend=c("h =", h), cex=0.8)
```



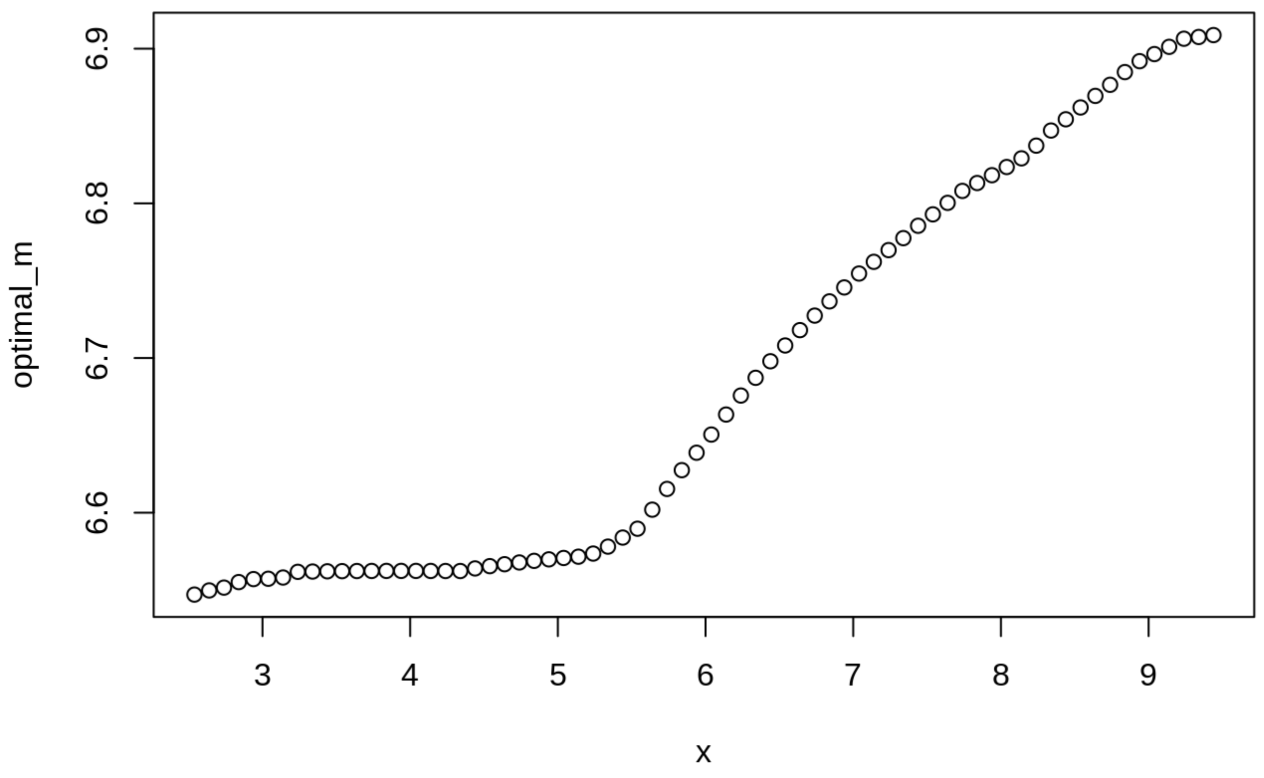
Question 9

```
# As recommended, we are going to take a small sample

set.seed(42) # We set seed(42) to uncover the mysteries of the universe
random_sample <- sample(1:nrow(df) , size = 100, replace = F)

# Data from our sample
log_density <- df$log_density[random_sample]
log_monthly_wages <- df$log_monthly_wages[random_sample]

# Following the instructions from question 4, we can construct
kernel_regression_one_less <- function(i, h){
  z <- (log_density[-i] - log_density[i])/h
  k <- epanechnikov(z)
  f_hat <- sum(k*log_monthly_wages[-i])/sum(k)
  return(f_hat)
}

# Cross-Validation function
cv <- function(h){
  i <- 1:100
  m_i <- sapply(i, function(x){kernel_regression_one_less(x,h)})
  # Some of the m_i turn out to be NaN. Therefore, i removed those from the vector
  i_nonzero <- setdiff(i, which(is.na(m_i)))
  m_i <- m_i[!is.na(m_i)]
  result <- (1/100)*sum((log_monthly_wages[i_nonzero] - m_i)^2)
  return(result)
}

# Recommended grid values
hgrid <- seq(0.1, 3, by=0.1)

# Estimation
y_cv_hat <- sapply(hgrid, cv)

# Minimun
y_cv_min <- min(y_cv_hat)

# h that minimizes my CV function on the recommended grid
h_optimal <- hgrid[which(y_cv_min == y_cv_hat)]
```

Question 10

Simple enough.

```
h <- h_optimal
optimal_m <- sapply(x, local_constant_estimator)
plot(x, optimal_m)
```


optimal_m
6.6
6.7
6.8
6.9
3
4
5
6
7
8
9
x