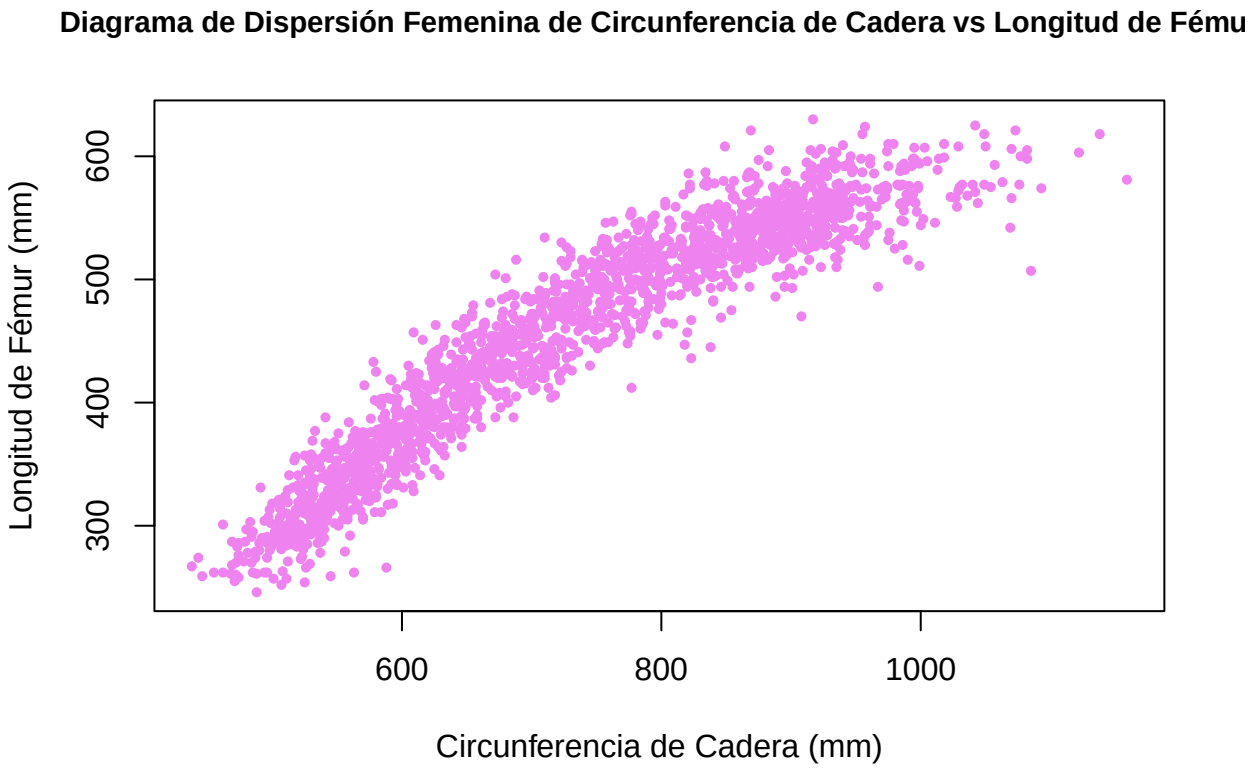
Regresión No Paramétrica

2024-07-13

Diagrama de dispersión de HIP.CIRCUMFERENCE(eje x) vs. BUTTOCK.KNEE.LENGTH (eje y) para las observaciones que corresponden al sexo femenino

```
datos_antropometricos <- read.csv("individuals.csv", sep = ";")
datos_femeninos <- datos_antropometricos[datos_antropometricos$SEX == 2,]
datos_fem_validos <- datos_femeninos[datos_femeninos$HIP.CIRCUMFERENCE != 0 & datos_femeninos$BUTTOCK.KN
attach(datos_fem_validos)
```

```
plot(HIP.CIRCUMFERENCE,BUTTOCK.KNEE.LENGTH,
     xlab = "Circunferencia de Cadera (mm)",
     ylab = "Longitud de Fémur (mm)",
     main = "Diagrama de Dispersión Femenina de Circunferencia de Cadera vs Longitud de Fémur",
     cex.main=0.9,
     col = "violet",
     pch = 16,
     cex = 0.7)
```



Se observa que hay una relación directamente proporcional entre las variables, de manera creciente: a mayor circunferencia de cadera, mayor longitud de fémur. Por otro lado, están distribuidos casi sin presencia de outliers (valores atípicos).

Interesa la distribución de la longitud del contorno de cadera de la población femenina en distintos grupos etários (o sea de edad), para lo cual consideraré la edad en meses registrada en AGE.IN.MONTHS. Teniendo en cuenta el primer, segundo y tercer cuartil de la variable AGE.IN.MONTHS formo 4 grupos etários y estimo la mediana de HIP.CIRCUMFERENCE en cada uno de ellos. Calculo un intervalo de confianza bootstrap normal de nivel 0.95 para cada una de las 4 medianas.

```
q1 <- quantile(AGE.IN.MONTHS, 0.25)
q2 <- quantile(AGE.IN.MONTHS, 0.5)
q3 <- quantile(AGE.IN.MONTHS, 0.75)

grupo_etario1 <- datos_fem_validos[AGE.IN.MONTHS <= q1,]
grupo_etario2 <- datos_fem_validos[AGE.IN.MONTHS > q1 & AGE.IN.MONTHS <= q2,]
grupo_etario3 <- datos_fem_validos[AGE.IN.MONTHS > q2 & AGE.IN.MONTHS <= q3,]
grupo_etario4 <- datos_fem_validos[AGE.IN.MONTHS > q3,]
```

```
mediana_grupo1 <- median(grupo_etario1$HIP.CIRCUMFERENCE)
mediana_grupo2 <- median(grupo_etario2$HIP.CIRCUMFERENCE)
```

```r
mediana_grupo3 <- median(grupo_etario3$HIP.CIRCUMFERENCE)
mediana_grupo4 <- median(grupo_etario4$HIP.CIRCUMFERENCE)
```

```r
mediana_grupo1
```

```
## [1] 556
```

```r
mediana_grupo2
```

```
## [1] 676
```

```r
mediana_grupo3
```

```
## [1] 799
```

```r
mediana_grupo4
```

```
## [1] 905
```

```r
estimar_se_mediana <-function(x, B = 1000){
  titahatboot <- rep(0, B)
  for(i in 1:B){
    Xboot <- sample(x, length(x), replace = TRUE)
    titahatboot[i] <- median(Xboot)
  }
  sqrt(mean((titahatboot-mean(titahatboot))**2))
}

intervalo_de_confianza <- function(x){
  set.seed(123)
  se_boot <- estimar_se_mediana(x)
  intervalo_boot <- c(median(x) - 1.96*se_boot,
                      median(x) + 1.96*se_boot)
}

intervalo1 <- c(intervalo_de_confianza(grupo_etario1$HIP.CIRCUMFERENCE))
intervalo2 <- c(intervalo_de_confianza(grupo_etario2$HIP.CIRCUMFERENCE))
intervalo3 <- c(intervalo_de_confianza(grupo_etario3$HIP.CIRCUMFERENCE))
intervalo4 <- c(intervalo_de_confianza(grupo_etario4$HIP.CIRCUMFERENCE))
```

```r
intervalo1
```

```
## [1] 550.6314 561.3686
```

```r
intervalo2
```

```
## [1] 668.8171 683.1829
```

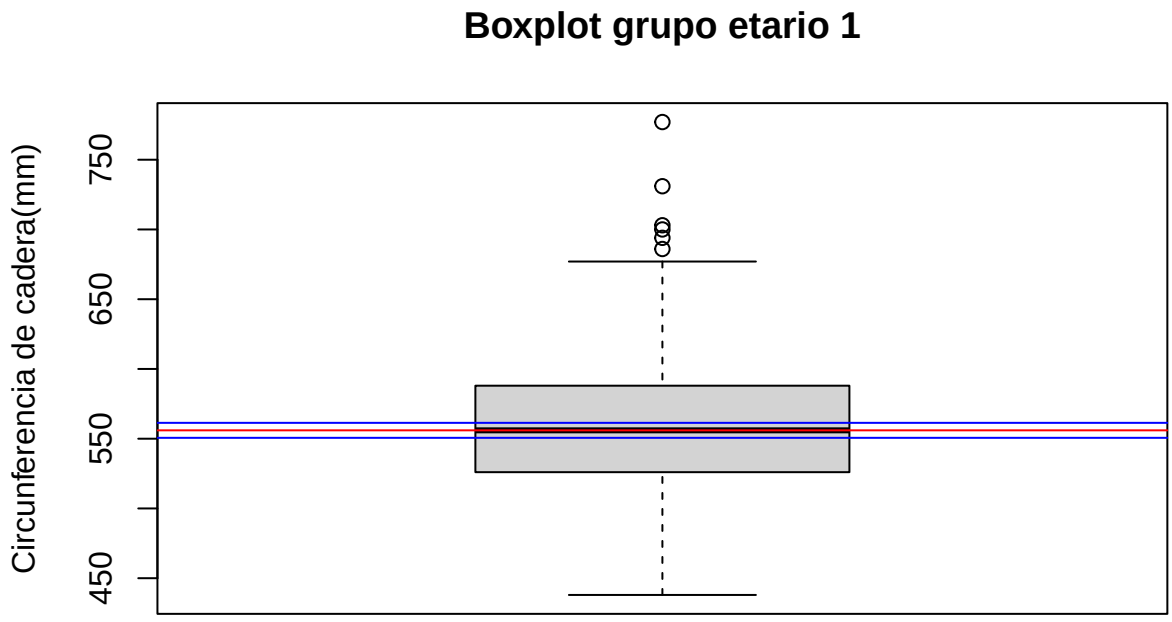
```r
intervalo3
```

```
## [1] 788.0066 809.9934
```

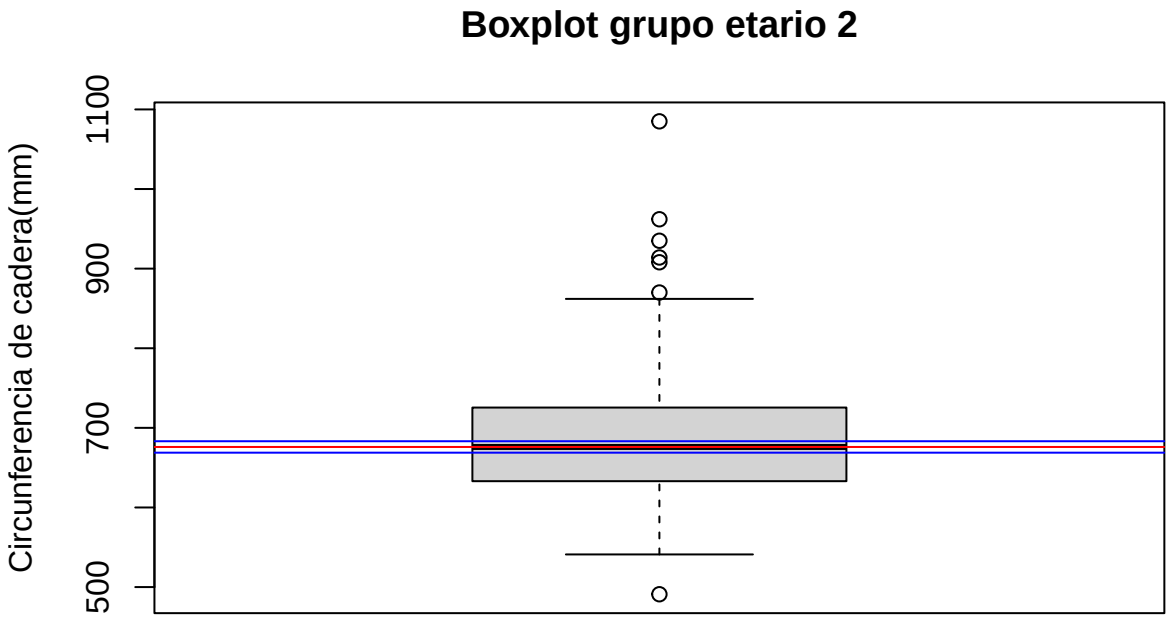
```r
intervalo4
```

```
## [1] 898.2562 911.7438
```

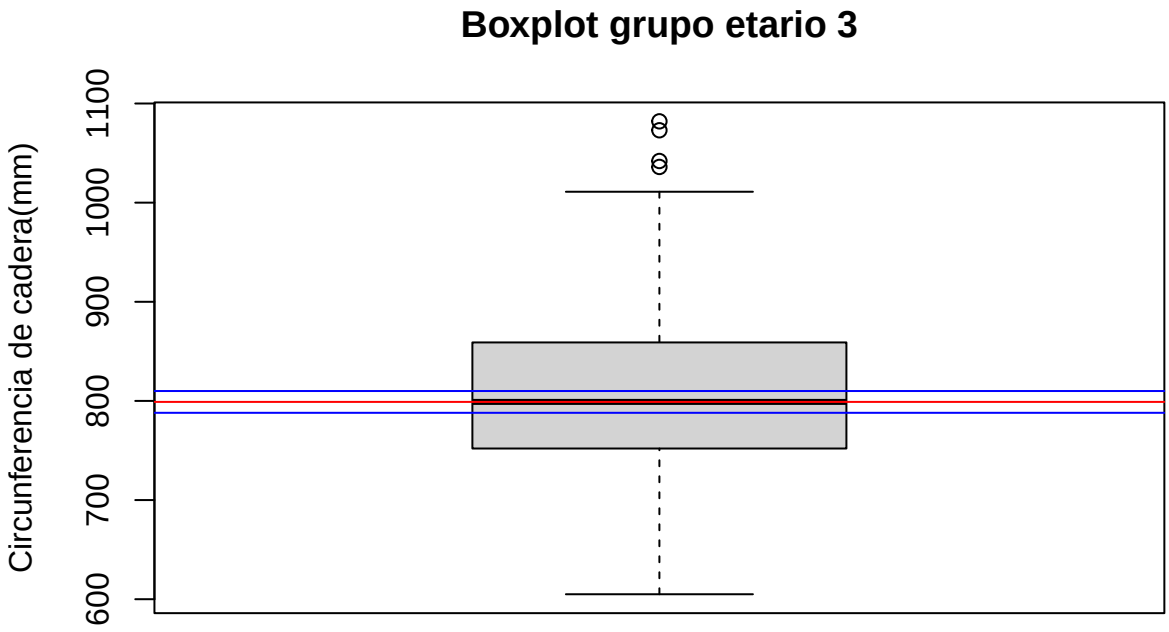
```r
boxplot(grupo_etario1$HIP.CIRCUMFERENCE,main = "Boxplot grupo etario 1", ylab= "Circunferencia de cadera
abline(h=mediana_grupo1, col="red")
abline(h= intervalo1[1],col= "blue")
abline(h= intervalo1[2],col= "blue")
```



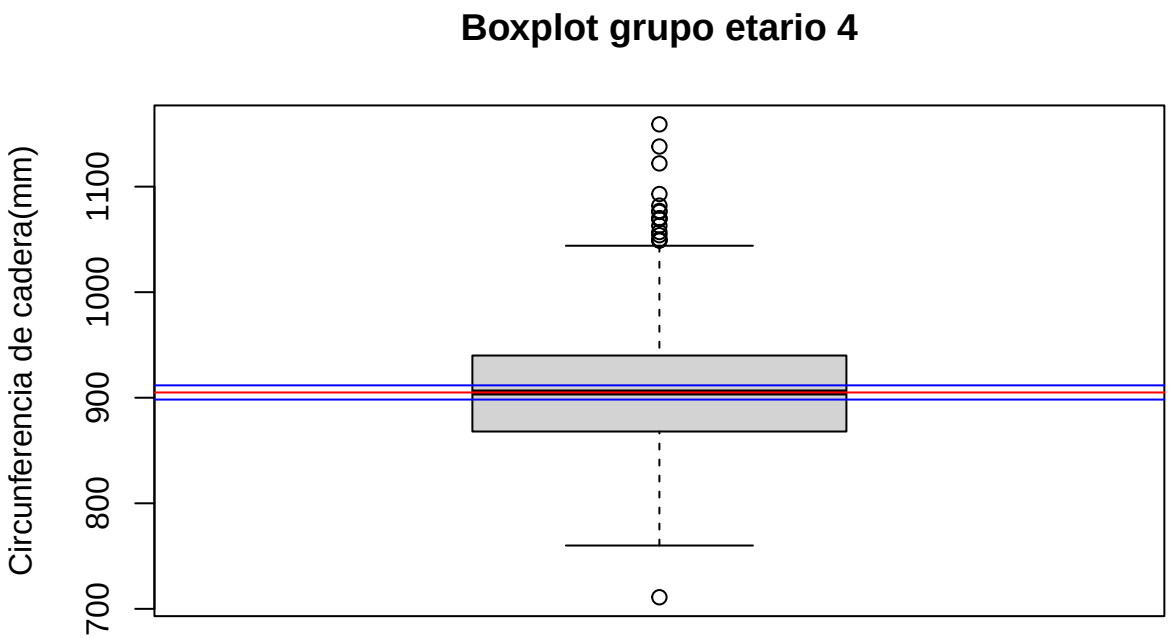

```r
boxplot(grupo_etario2$HIP.CIRCUMFERENCE, main = "Boxplot grupo etario 2", ylab= "Circunferencia de cade
abline(h=mediana_grupo2, col="red")
abline(h= intervalo2[1],col= "blue")
abline(h= intervalo2[2],col= "blue")
```




```
boxplot(grupo_etario3$HIP.CIRCUMFERENCE, main = "Boxplot grupo etario 3", ylab= "Circunferencia de cade
abline(h=mediana_grupo3, col="red")
abline(h= intervalo3[1],col= "blue")
abline(h= intervalo3[2],col= "blue")
```




```
boxplot(grupo_etario4$HIP.CIRCUMFERENCE, main ="Boxplot grupo etario 4", ylab= "Circunferencia de cadera
abline(h=mediana_grupo4, col="red")
abline(h= intervalo4[1],col= "blue")
abline(h= intervalo4[2],col= "blue")
```



Obtuve el intervalo de confianza utilizando el método de bootstrap. Para ello, genero muestras con reemplazo y calculo las medianas y su error estándar. Con estos datos genero los intervalos de confianza de nivel 0.95 con la fórmula para mediana ± 1.96 * error estándar vista en clase. En todo el proceso asumo la distribución normal de las medianas.

Observando los resultados obtenidos, noto que la mediana, en todos los casos, parece coincidir con la media y además el intervalo de confianza, no solo contiene a la mediana estimada, sino también a la media.

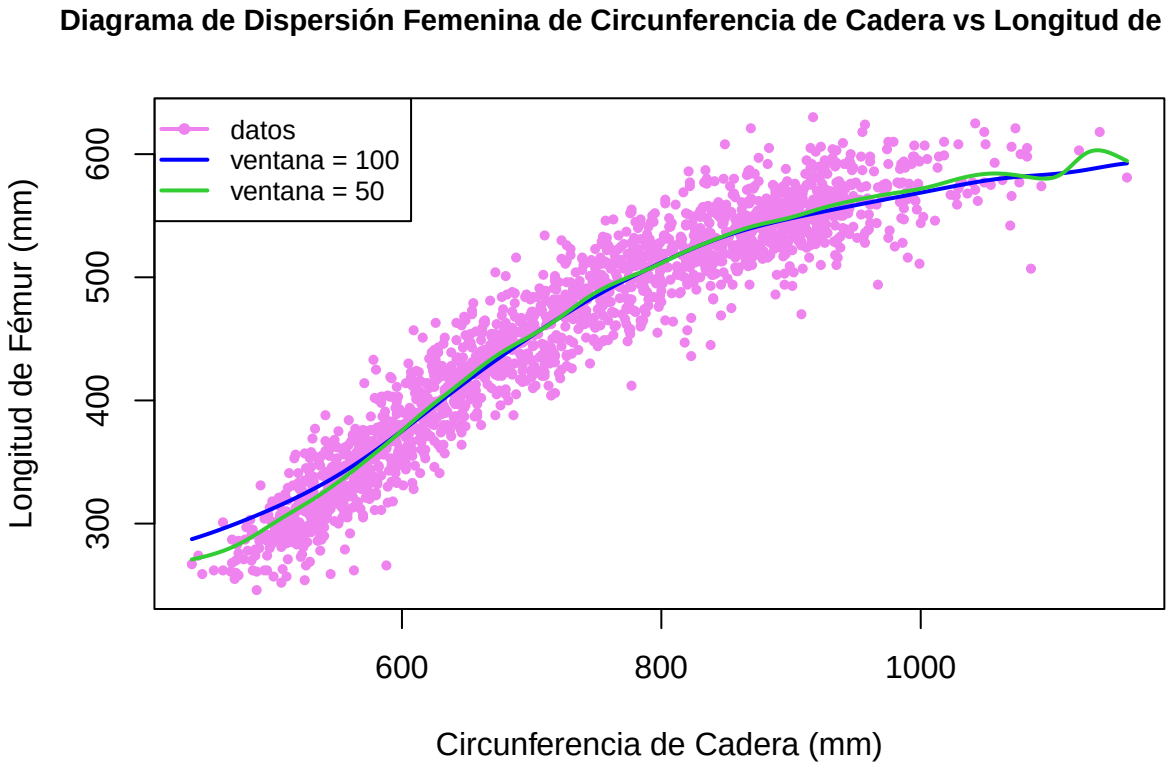
Evaluao un ajuste de regresión para las variables HIP.CIRCUMFERENCE (x) y BUTTOCK.KNEE.LENGTH (y)

Ajuste normal con ventana 50 y 100

```
ajuste_normal_100 <- ksmooth(HIP.CIRCUMFERENCE, BUTTOCK.KNEE.LENGTH, kernel = "normal", bandwidth = 100)
ajuste_normal_50 <- ksmooth(HIP.CIRCUMFERENCE, BUTTOCK.KNEE.LENGTH, kernel = "normal", bandwidth = 50)
```

```
plot(HIP.CIRCUMFERENCE,BUTTOCK.KNEE.LENGTH,
     xlab = "Circunferencia de Cadera (mm)",
     ylab = "Longitud de Fémur (mm)",
     main = "Diagrama de Dispersión Femenina de Circunferencia de Cadera vs Longitud de Fémur",
     cex.main=0.9,
     col = "violet",
     pch = 16,
     cex = 0.7)
```

```r
lines(ajuste_normal_100$x, ajuste_normal_100$y, col = "blue", lwd = 2)
lines(ajuste_normal_50$x, ajuste_normal_50$y, col = "limegreen", lwd = 2)

legend("topleft", legend = c("datos","ventana = 100", "ventana = 50"), col = c("violet", "blue", "limeg
```



La curva de ajuste correspondiente a la ventana más grande (100) es más suave. Por otro lado, la que posee menor ventana (50) tiene pequeñas fluctuaciones, sobre todo para valores altos en ambas variables de interés, capturando así, más variaciones en dicha aŕea. Sin embargo, son relativamente pocos los datos en esa zona. Si quisiera estudiar un área específica de la circunferencia me inclinaría por 50. Pero como quiero algo más general, resulta mejor la de 100.

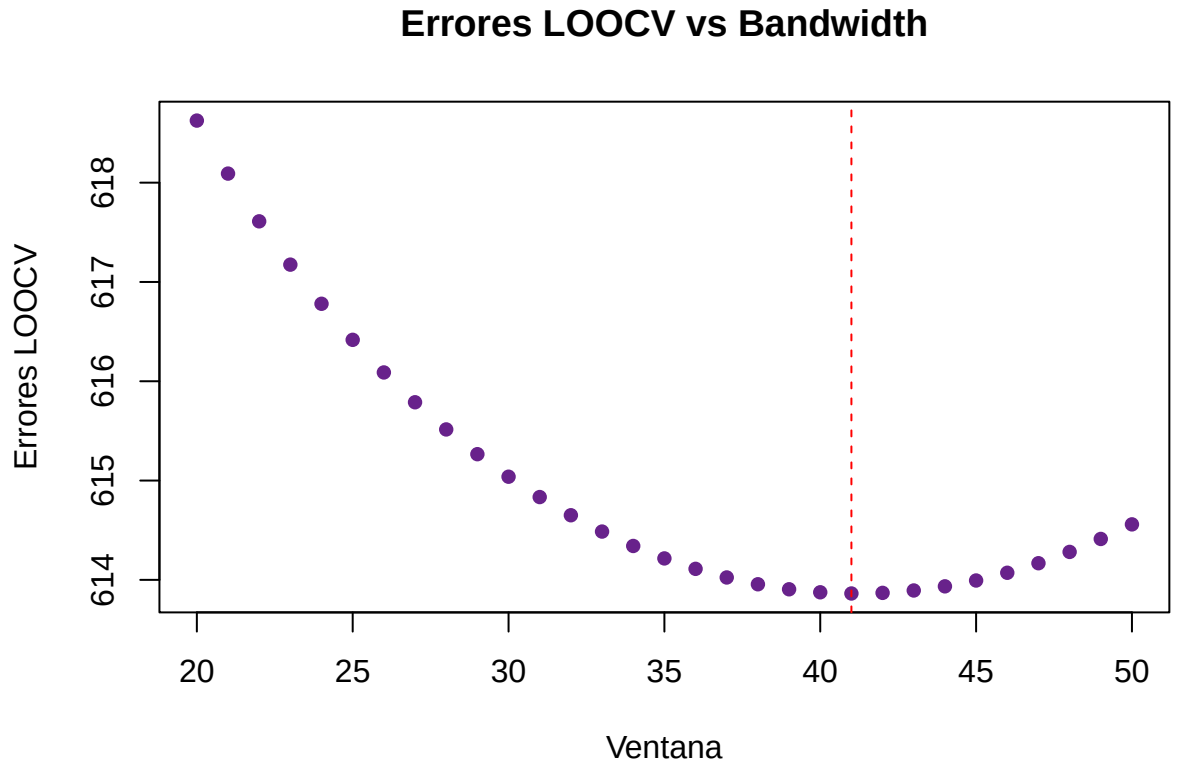
Búsqueda de la ventana óptima para ksmooth con núcleo normal para el parámetro bandwidth, con el criterio de convalidación cruzada basado en leave-one-out y búsqueda en una grilla de bandwidth entre 20 y 50 con paso 1

```r
loocv_bw <- function(x,y,bw){
  errores <- numeric(length(x))
  for(i in 1: length(x)){
    # saco elem i
    x_i = x[-i]
    y_i = y[-i]
    # calculo la estimación
    pred_ks <- ksmooth(x_i,y_i, kernel = "normal", bandwidth=bw, x.point= x[i])
```

```
    # calculo error cuadrático de convalidación cruzada:
    errores[i] <- (y[i]-pred_ks$y)**2
  }
  return(mean(errores))
}

set.seed(34)
grilla_bw <- seq(20, 50, by = 1)
loocv_errores <- sapply(grilla_bw, loocv_bw, x = HIP.CIRCUMFERENCE, y = BUTTOCK.KNEE.LENGTH)

mejor_bw <- grilla_bw[which.min(loocv_errores)]
mejor_bw
```

```
## [1] 41
```

```
plot(grilla_bw, loocv_errores,
     main = "Errores LOOCV vs Bandwidth",
     xlab = "Ventana",
     ylab = "Errores LOOCV",
     col = "darkorchid4",
     pch = 16,
     cex = 1)
abline(v = mejor_bw, col = "red", lty = 2)
```



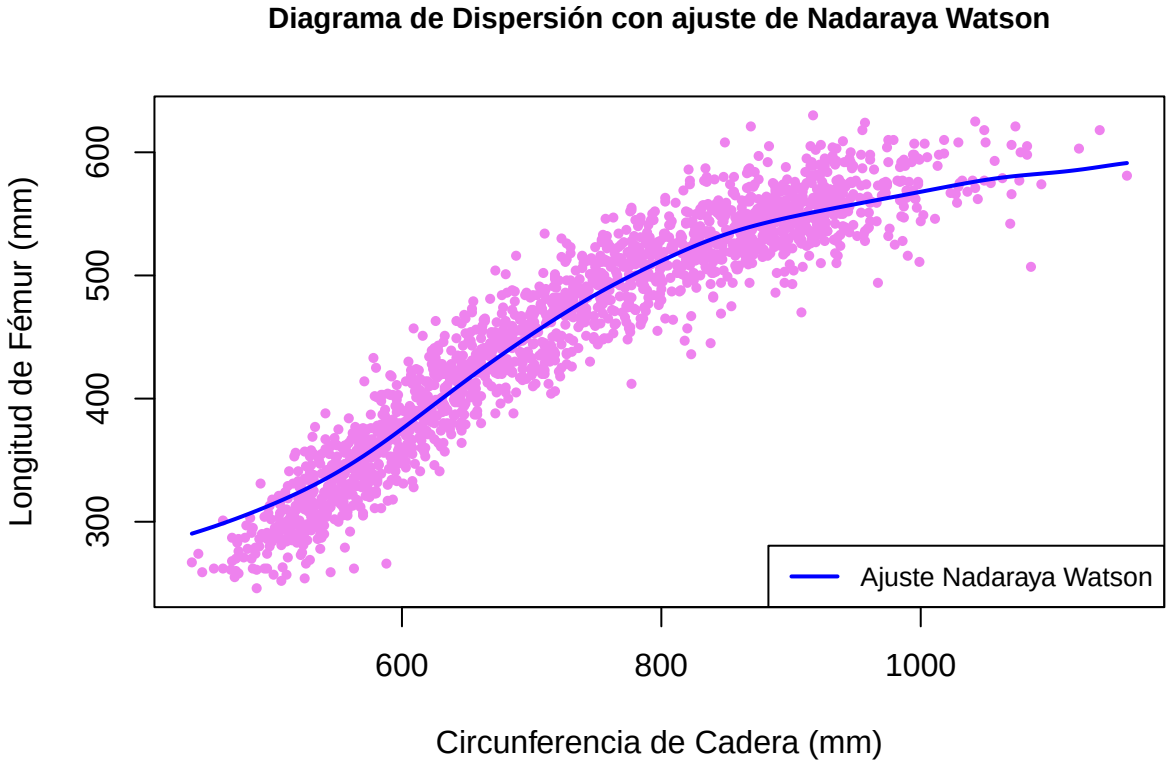
Análisis estimador de Nadaraya–Watson

```
nwsmooth <- function(x, y, xi, bw) {
  nucleo_normal_xi <- dnorm((xi - x) / bw)
  suma_nucleos <- sum(nucleo_normal_xi * y)
  suma_nucleos_denominador <- sum(nucleo_normal_xi)
  estimador_NW <- suma_nucleos / suma_nucleos_denominador
  return(estimador_NW)
}

loocv_bw_NW <- function(x,y,bw){
  errores_NW <- numeric(length(x))
  for(i in 1: length(x)){
    # saco elem i
    x_i = x[-i]
    y_i = y[-i]
    # calculo la estimación
    nucleo_NW <- nwsmooth(x_i,y_i,x[i],bw)

    # calculo error cuadrático de convalidación cruzada:
    errores_NW[i] <- (y[i]-nucleo_NW)**2
  }
  return(mean(errores_NW))
}

grilla_bw_NW <- seq(10, 50, by = 1)
loocv_errores_NW <- sapply(grilla_bw_NW, loocv_bw_NW , x = HIP.CIRCUMFERENCE, y = BUTTOCK.KNEE.LENGTH)
mejor_bw_NW <- grilla_bw_NW[which.min(loocv_errores_NW)]
mejor_bw_NW
```

```
## [1] 15
```

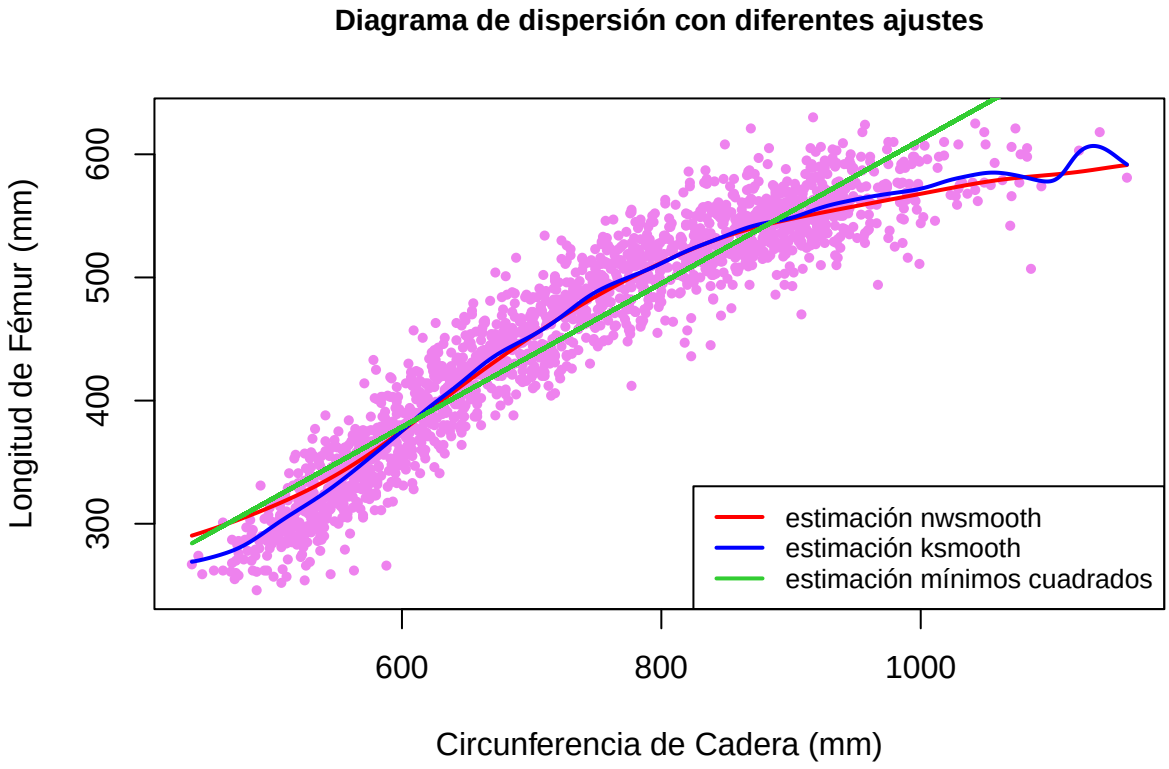
Para este caso, en primera instancia mantuve la grilla del item anterior. Sin embargo, como el valor de la ventana óptima resultó ser 20, decidi emplear una grilla nueva (entre 10 y 50) para encontrar la ventana óptima.

```
estimador_NW_bw_optimo<- sapply(HIP.CIRCUMFERENCE, nwsmooth,HIP.CIRCUMFERENCE,BUTTOCK.KNEE.LENGTH,mejor

plot(HIP.CIRCUMFERENCE,BUTTOCK.KNEE.LENGTH,
     xlab = "Circunferencia de Cadera (mm)",
     ylab = "Longitud de Fémur (mm)",
     main = "Diagrama de Dispersión con ajuste de Nadaraya Watson",
     cex.main=0.9,
     col = "violet",
     pch = 16,
     cex = 0.7)

x_seq <- seq(min(HIP.CIRCUMFERENCE), max(HIP.CIRCUMFERENCE), length.out = 300)

y_est <- sapply(x_seq, nwsmooth, x = HIP.CIRCUMFERENCE, y = BUTTOCK.KNEE.LENGTH, bw = 40)
lines(x_seq, y_est, col = "blue", lwd = 2)

legend("bottomright", legend = c("Ajuste Nadaraya Watson"), col = "blue", lwd = 2, cex=0.8)
```



Análisis de la estimación dela regresión no paramétrica de la ventana óptima, comparado con la implementación de nwsmooth como con ksmooth. Asimismo, superpongo la recta que obtiene utilizando el método de mínimos cuadrados.

```
y_est_ks <- ksmooth(HIP.CIRCUMFERENCE,BUTTOCK.KNEE.LENGTH, kernel = "normal", bandwidth= mejor_bw)
modelo_lm <- lm(BUTTOCK.KNEE.LENGTH ~ HIP.CIRCUMFERENCE)

y_sombrero_lm <- predict(modelo_lm)

plot(HIP.CIRCUMFERENCE,BUTTOCK.KNEE.LENGTH,
     xlab = "Circunferencia de Cadera (mm)",
     ylab = "Longitud de Fémur (mm)",
     main = "Diagrama de dispersión con diferentes ajustes",
     cex.main=0.9,
     col = "violet",
     pch = 16,
     cex = 0.7)

lines(x_seq, y_est, col = "red", lwd = 2)
lines(y_est_ks$x, y_est_ks$y ,col = "blue", lwd = 2)
lines(HIP.CIRCUMFERENCE, y_sombrero_lm, col= "limegreen", lwd=2)
legend("bottomright", legend = c("estimación nwsmooth", "estimación ksmooth", "estimación mínimos cuadra
```



Al observar el gráfico, noto que la recta de ajuste por mínimos cuadrados no se ajusta de la manera más precisa a los datos. Por otro lado, si bien el estimador "ksmooth" captura mejor la distribución de los datos, presenta mayor variabilidad (es muy sensible a datos atípicos, como se ve en el caso de mayores valores en las variables) en comparación al ajuste realizado con "nwsmooth". Dicho esto, si priorizáramos tener una estimación más suave elegiríamos el ajuste "nwsmooth".

Estimador con la función linearsmooth

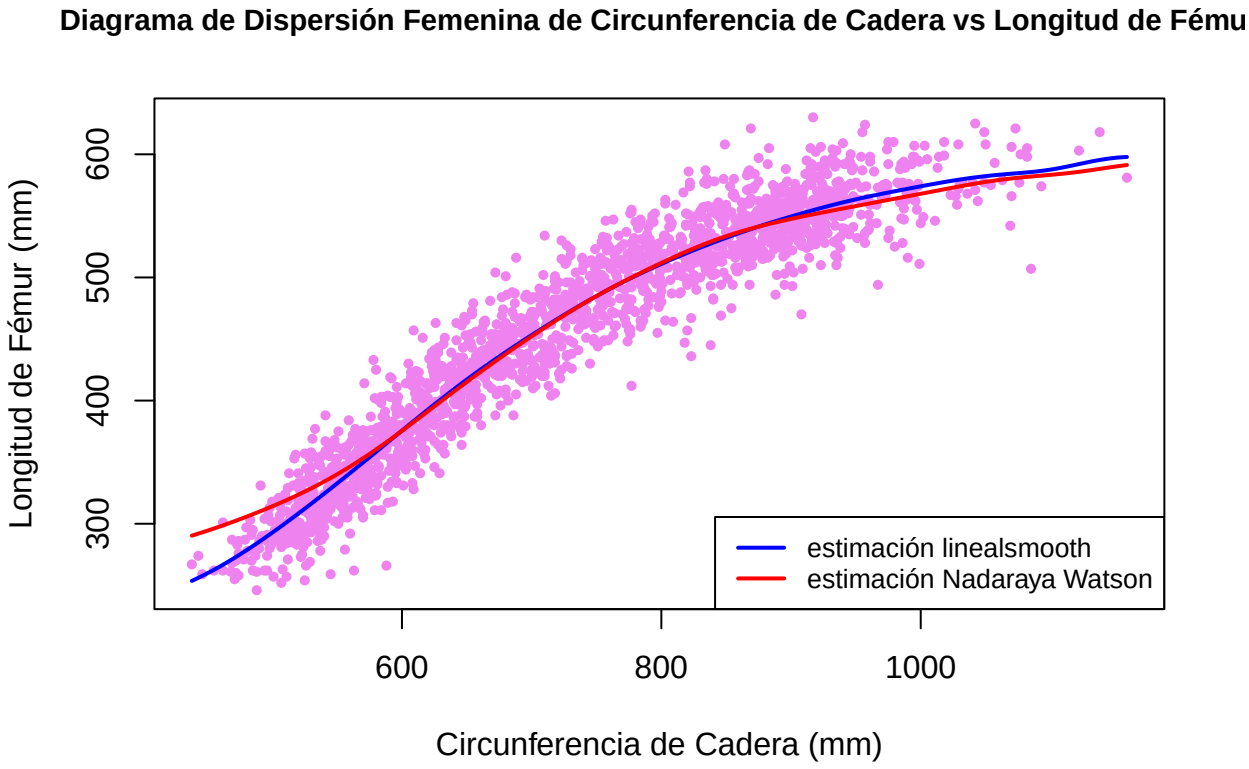
```
linearsmooth <- function(x, y, x0, h) {
  # calculo W la matriz diagonal
  W <- dnorm((x - x0) / h)  # usamos el núcleo normal
  W_matriz_diagonal <- diag(W / sum(W))

  # creo la matriz X
  X <- cbind(1, x - x0)

  # obtengo matricialmente los estimadores
  XtWX_inversa <- solve(t(X) %*% W_matriz_diagonal %*% X)
  XtWY <- t(X) %*% (W_matriz_diagonal %*% y)
  return(XtWX_inversa %*% XtWY)
}
```

```
# aplico el estimador a los datos:
x <- HIP.CIRCUMFERENCE
```

```r
y <- BUTTOCK.KNEE.LENGTH
ventana <- 40

x_seq <- seq(min(x), max(x), length.out = 300)

# calculo los valores estimados de y según cada x:
y_estimados <- sapply(x_seq, function(x0) {
  coeficientes <- linearsmooth(x, y, x0, 40)
  return (coeficientes[1]) # por ser el estimador local lineal el coeficiente a0 (fórmula sección (6) d
})
```

Diagrama de Dispersión Femenina de Circunferencia de Cadera vs Longitud de Fému

```r
plot(HIP.CIRCUMFERENCE,BUTTOCK.KNEE.LENGTH,
     xlab = "Circunferencia de Cadera (mm)",
     ylab = "Longitud de Fémur (mm)",
     main = "Diagrama de Dispersión Femenina de Circunferencia de Cadera vs Longitud de Fémur",
     cex.main=0.9,
     col = "violet",
     pch = 16,
     cex = 0.7)

lines(x_seq, y_estimados, col = "blue", lwd = 2)
lines(x_seq, y_est, col = "red", lwd = 2)
legend("bottomright", legend = c("estimación linealsmooth", "estimación Nadaraya Watson"), col = c( "blu
```



Si bien ambos ajustes son muy similares, en este caso difieren para los valores más bajos de las variables, ajustando mejor la estimación hecha por "linealsmooth".