Métodos no paramétricos

Ejercicios/prácticas propuestas

Borja Souto Prego

2025-01-02

Parte A: Métodos de distribución libre

Ejercicio 1 En este ejercicio, donde se pide evaluar la conducta de las pruebas de bondad de ajuste de Kolmogorov-Smirnov, Cramér-von Mises y Anderson-Darling para detectar diferentes desviaciones de una normal estándar, consideraremos el caso (B) N(0, σ), con un valor de σ=0.48.

```
library(Hmisc)
```

```
##
## Adjuntando el paquete: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(goftest)
```

Aprovechando que se solicita en el apartado (d), elaboramos una función (`evaluar_pruebas`) que realiza los apartados (a), (b) y (c) para (n,θ=σ,α) arbitrarios. Es decir:

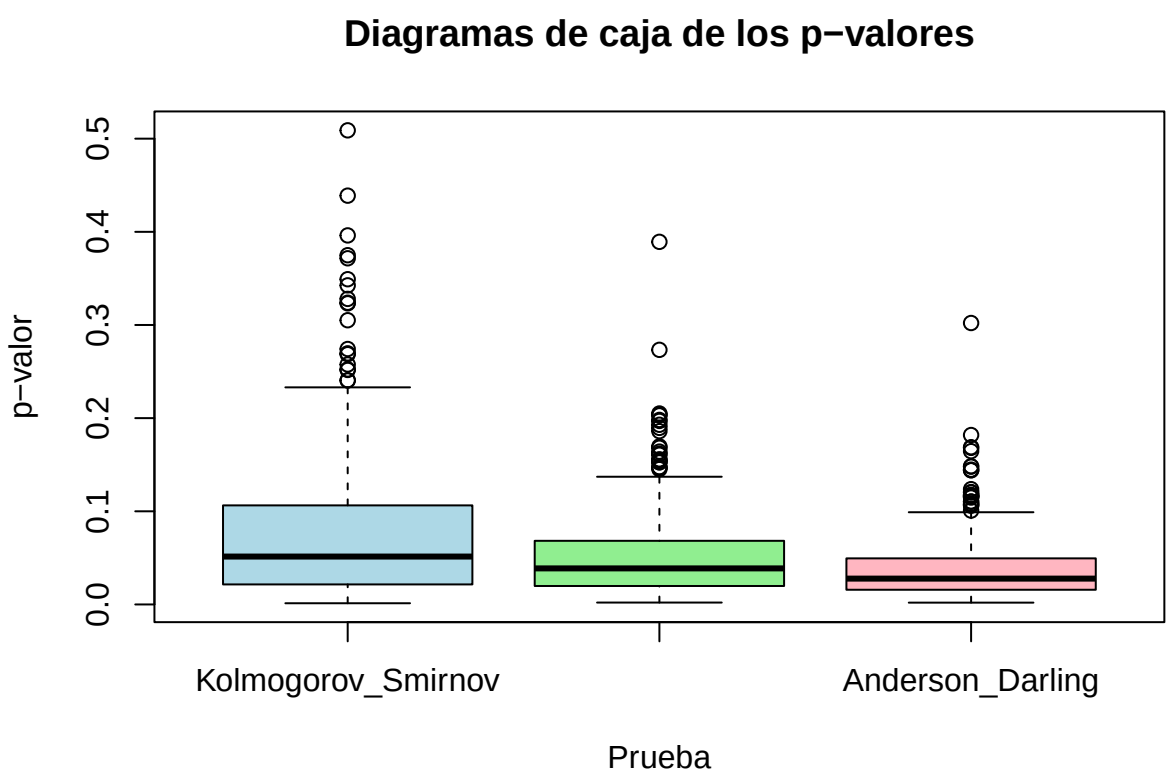
- Genera 500 m.a.s. de tamaño n=30 de la distribución N(0,0.48)
- Evalúa los tres estadísticos para cada muestra y calcula la frecuencia relativa de rechazos al nivel de significación $\alpha = 0.05$.
- Para los p-valores obtenidos en cada prueba: genera diagramas de caja, histogramas, gráficos enfrentando la distribución empírica a la teórica y estadísticos descriptivos.

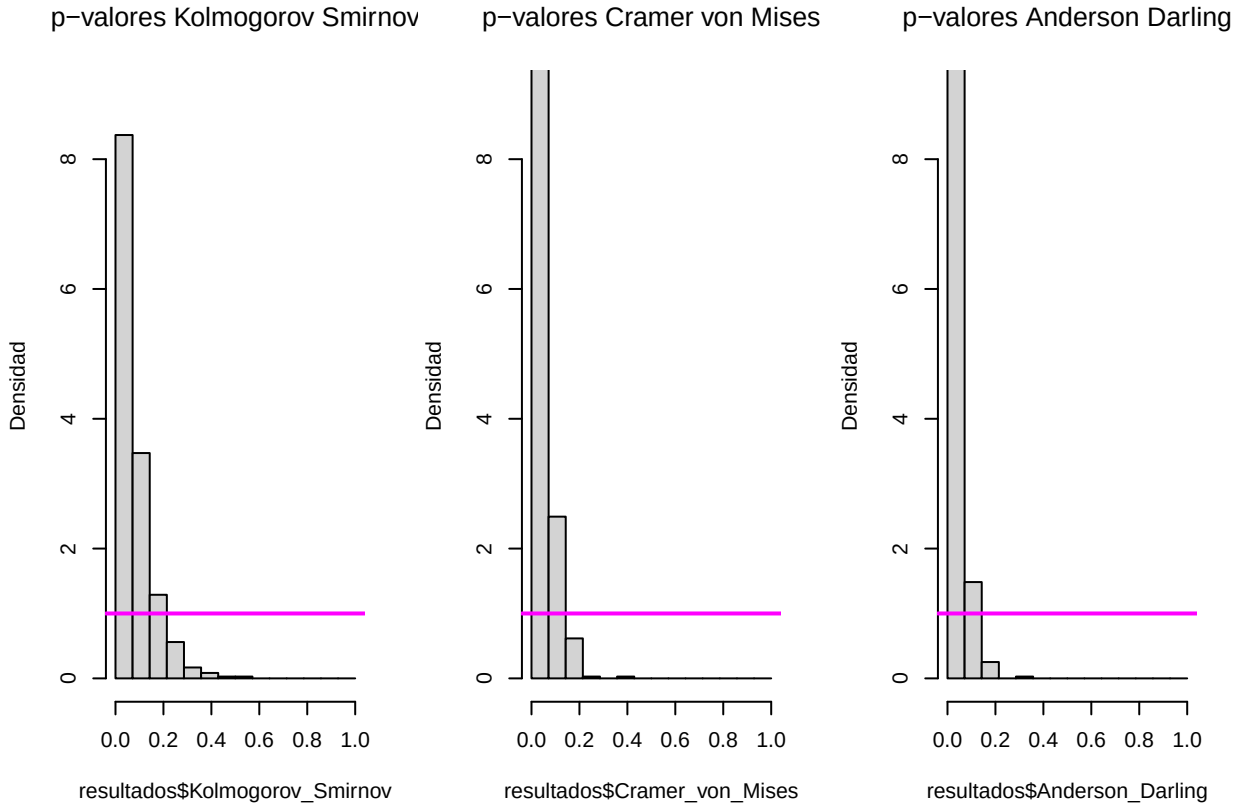
```
evaluar_pruebas <- function(n, theta, alpha, T = 500) {
  # Generar T=500 muestras de tamaño n de la distribución N(0,0.48)
  set.seed(13000)
  muestras <- replicate(T, rnorm(n, mean = 0, sd = theta))

  resultados <- data.frame(
    Kolmogorov_Smirnov = numeric(T),
    Cramer_von_Mises = numeric(T),
    Anderson_Darling = numeric(T)
  )
```

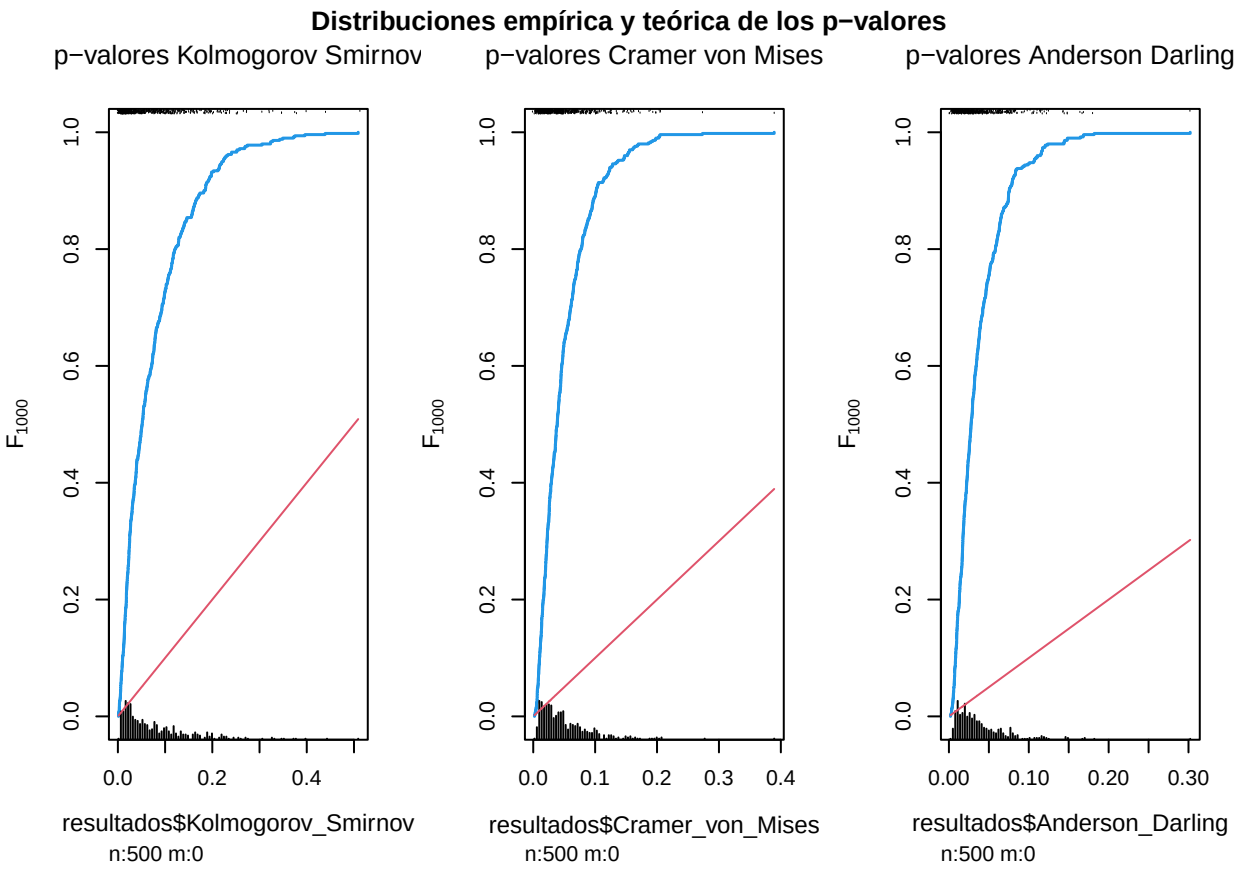
```r
for (i in 1:T) {
  muestra <- muestras[, i]

  # Prueba de Kolmogorov-Smirnov
  ks_test <- ks.test(x=muestra, y="pnorm")
  resultados$Kolmogorov_Smirnov[i] <- ks_test$p.value

  # Prueba de Cramér-von Mises
  cvm_test <- goftest::cvm.test(x=muestra, null="pnorm")
  resultados$Cramer_von_Mises[i] <- cvm_test$p.value

  # Prueba de Anderson-Darling
  ad_test <- goftest::ad.test(x=muestra, null="pnorm")
  resultados$Anderson_Darling[i] <- ad_test$p.value
}

# Frecuencia relativa de rechazos
rechazos <- colMeans(resultados < alpha)

print("Frecuencia relativa de rechazos:")
print(rechazos)
print("Resumen descriptivo de los p-valores:")
print(summary(resultados))

# Gráficos
resultados_long <- reshape2::melt(resultados)

# Diagrama de caja con base R
boxplot(value ~ variable, data = resultados_long,
        main = "Diagramas de caja de los p-valores",
        xlab = "Prueba", ylab = "p-valor",
        col = c("lightblue", "lightgreen", "lightpink"))

par(mfrow = c(1,3))
hist(resultados$Kolmogorov_Smirnov, breaks=seq(0,1,l=15), probability=TRUE, ylim=c(0,9),
     ylab="Densidad", main = expression(paste("p-valores Kolmogorov Smirnov")))
abline(h = 1, col = "magenta", lwd=2)

hist(resultados$Cramer_von_Mises, breaks=seq(0,1,l=15), probability=TRUE, ylim=c(0,9),
     ylab="Densidad", main = expression(paste("p-valores Cramer von Mises")))
abline(h = 1, col = "magenta", lwd=2)

hist(resultados$Anderson_Darling, breaks=seq(0,1,l=15), probability=TRUE, ylim=c(0,9),
     ylab="Densidad", main = expression(paste("p-valores Anderson Darling")))
abline(h = 1, col = "magenta", lwd=2)

title("Histogramas de los p-valores de las diferentes pruebas", line = -1, outer = TRUE)

par(mfrow = c(1,3))
CDF_KS <- Ecdf(resultados$Kolmogorov_Smirnov, main=expression(paste("p-valores Kolmogorov Smirnov")),
               ylab=expression(F[1000]), cex.lab=1.1, col=4, lwd=1.5)
scat1d(resultados$Kolmogorov_Smirnov)
histSpike(resultados$Kolmogorov_Smirnov, add=TRUE, frac=.15)
```

```r
  lines(CDF_KS$x,punif(CDF_KS$x),col=2)

  CDF_CVM <- Ecdf(resultados$Cramer_von_Mises, main=expression(paste("p-valores Cramer von Mises")),
                  ylab=expression(F[1000]), cex.lab=1.1, col=4, lwd=1.5)
  scat1d(resultados$Cramer_von_Mises)
  histSpike(resultados$Cramer_von_Mises, add=TRUE, frac=.15)
  lines(CDF_CVM$x,punif(CDF_CVM$x),col=2)

  CDF_AD <- Ecdf(resultados$Anderson_Darling, main=expression(paste("p-valores Anderson Darling")),
                   ylab=expression(F[1000]), cex.lab=1.1, col=4, lwd=1.5)
  scat1d(resultados$Anderson_Darling)
  histSpike(resultados$Anderson_Darling, add=TRUE, frac=.15)
  lines(CDF_AD$x,punif(CDF_AD$x),col=2)
  title("Distribuciones empírica y teórica de los p-valores", line = -1, outer = TRUE)
}

# Llamada a la función para (N(0, theta)) con theta = 0.48
evaluar_pruebas(n = 30, theta = 0.48, alpha = 0.05)
```

```
## [1] "Frecuencia relativa de rechazos:"
## Kolmogorov_Smirnov   Cramer_von_Mises   Anderson_Darling
##              0.492              0.640              0.754
## [1] "Resumen descriptivo de los p-valores:"
##  Kolmogorov_Smirnov Cramer_von_Mises   Anderson_Darling
##  Min.   :0.001306   Min.   :0.002043   Min.   :0.001962
##  1st Qu.:0.021509   1st Qu.:0.019903   1st Qu.:0.015790
##  Median :0.051406   Median :0.038701   Median :0.027732
##  Mean   :0.076036   Mean   :0.050147   Mean   :0.037213
##  3rd Qu.:0.106267   3rd Qu.:0.068182   3rd Qu.:0.049449
##  Max.   :0.508927   Max.   :0.389263   Max.   :0.302108

## No id variables; using all as measure variables
```


Diagramas de caja de los p−valores
p−valor
0.0
0.1
0.2
0.3
0.4
0.5
Kolmogorov_Smirnov
Anderson_Darling
Prueba


Histogramas de los p−valores de las diferentes pruebas
p−valores Kolmogorov Smirnov
p−valores Cramer von Mises
p−valores Anderson Darling
Densidad
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
1.0
resultados$Kolmogorov_Smirnov
resultados$Cramer_von_Mises
resultados$Anderson_Darling



Como podemos observar, la frecuencia relativa de rechazos en las 3 pruebas es muy alta (0.492, 0.640 y 0.754). La que mejores resultados ha proporcionado es, por lo tanto, la prueba de Anderson Darling, ya que es la que más rechazos ha realizado (75.4%). En los diagramas de cajas vemos que los p-valores en las 3 pruebas toman valores muy pequeños, principalmente entre valores muy cercanos a 0.0 y 0.1. Esto se puede observar también en los histogramas, donde además, vemos que no se distribuyen siguiendo una U(0,1). Las gráficas que enfrentan la distribución empírica y la teórica de los p-valores confirman de nuevo que los p-valores no siguen una U(0,1).

Los resultados obtenidos son coherentes con la prueba que se está a realizar, donde tenemos muestras de una N(0,0.48), y la hipótesis nula es que siguen una N(0,1), por lo que es lógico tener muchos p-valores pequeños (y muchos rechazos), y por lo tanto, que los p-valores no se distribuyan en torno a una U(0,1).

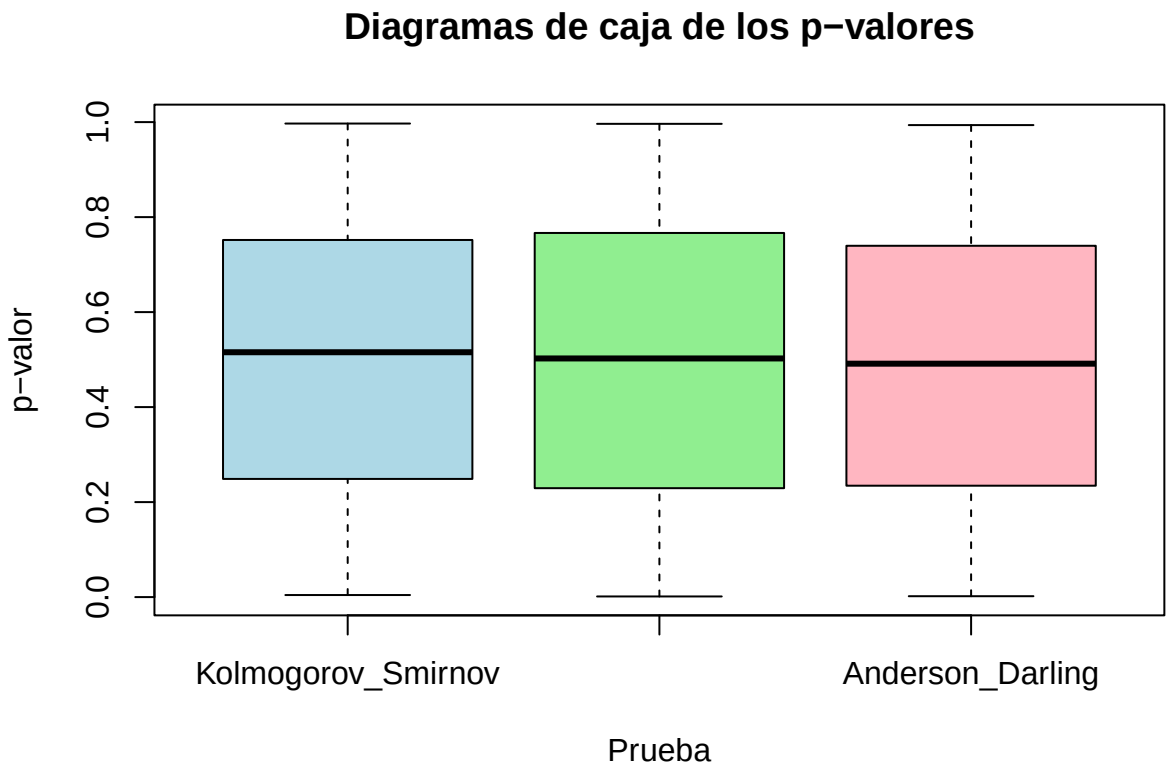
Para la realización del apartado (e) se utiliza de nuevo la función `evaluar_pruebas`, pero en este caso con el valor θ=1, para generar 500 m.a.s. de tamaño n=30 de una N(0,1).

```
evaluar_pruebas(n = 30, theta = 1, alpha = 0.05)
```

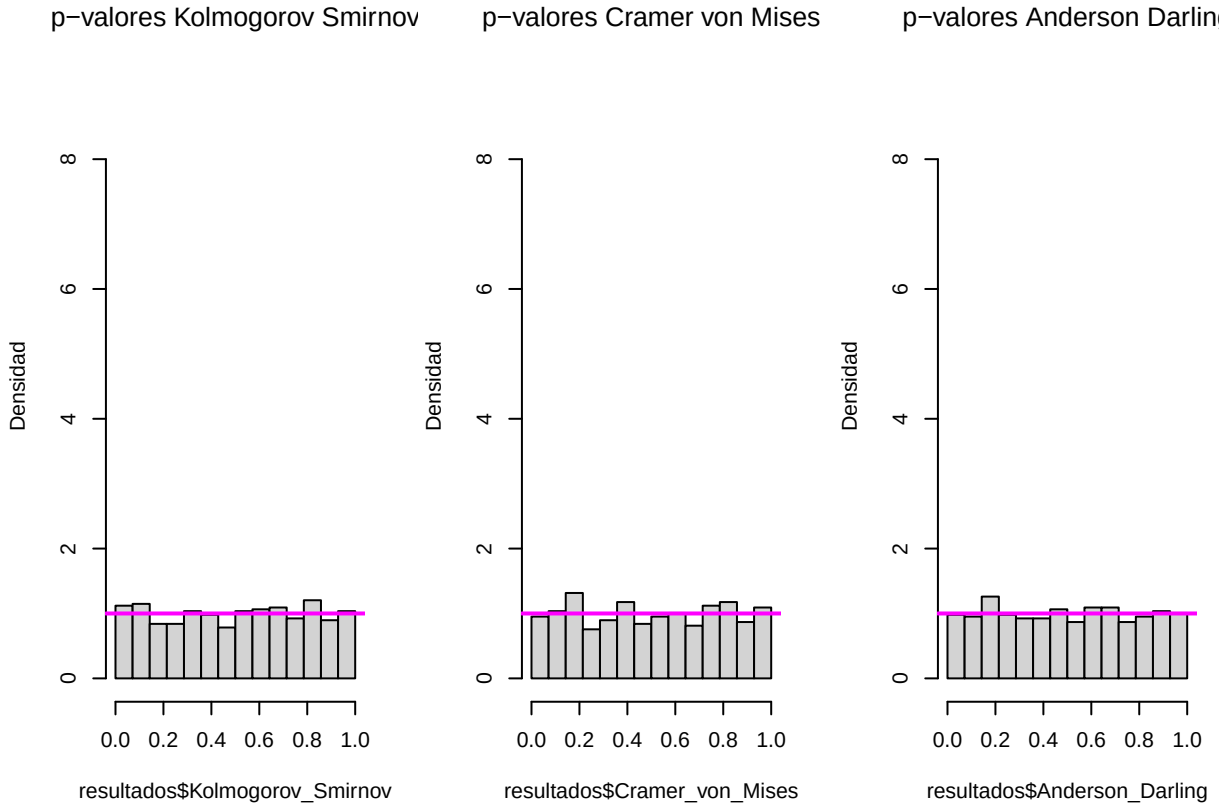
```
## [1] "Frecuencia relativa de rechazos:"
## Kolmogorov_Smirnov   Cramer_von_Mises   Anderson_Darling
##              0.058              0.054              0.054
## [1] "Resumen descriptivo de los p-valores:"
##  Kolmogorov_Smirnov Cramer_von_Mises   Anderson_Darling
##  Min.   :0.004362   Min.   :0.001379   Min.   :0.00186
##  1st Qu.:0.248846   1st Qu.:0.230412   1st Qu.:0.23502
##  Median :0.515484   Median :0.502417   Median :0.49141
##  Mean   :0.501540   Mean   :0.501264   Mean   :0.49783
##  3rd Qu.:0.751607   3rd Qu.:0.765714   3rd Qu.:0.73933
```

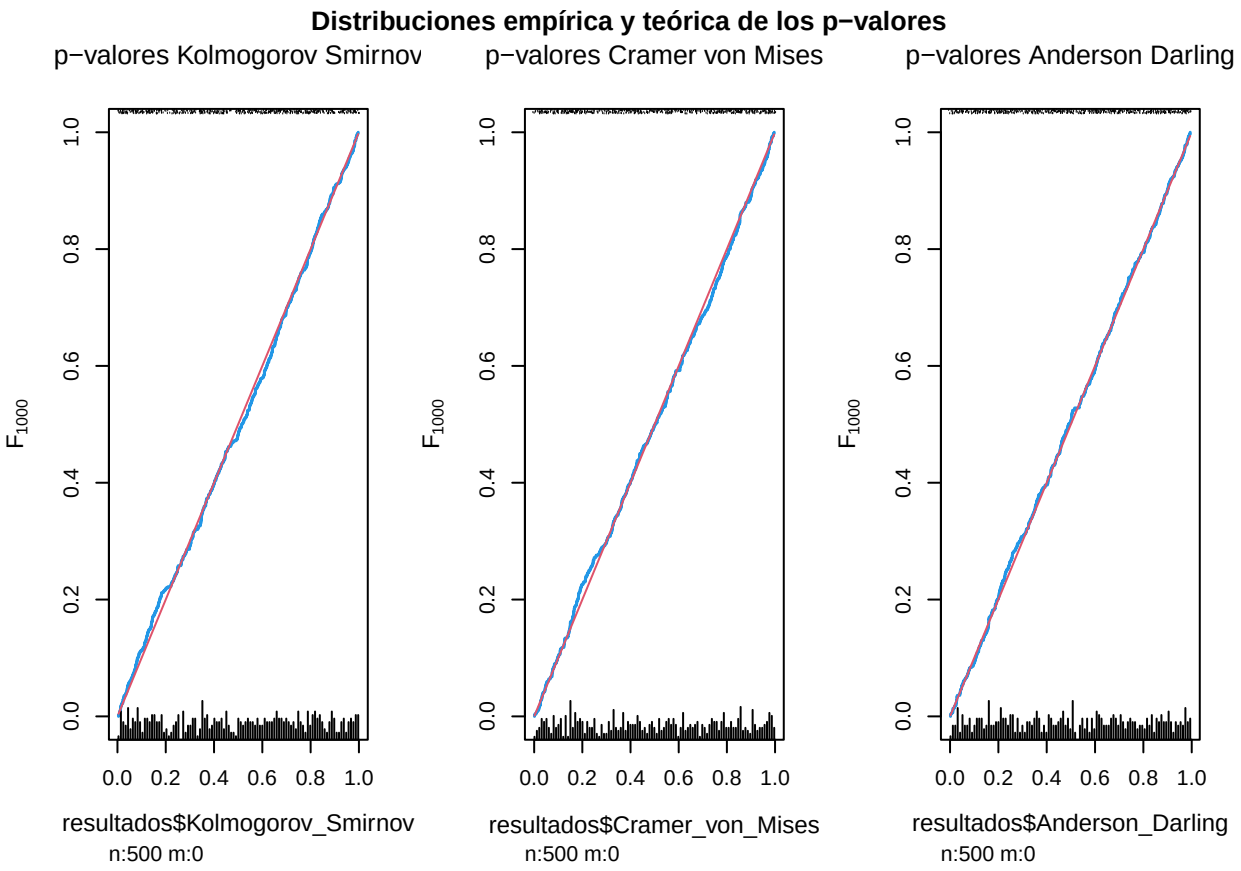
```
##  Max.   :0.996964   Max.   :0.996378   Max.   :0.99372
```

```
## No id variables; using all as measure variables
```



Histogramas de los p−valores de las diferentes pruebas





Como era de esperar, la frecuencia relativa de rechazos, que coincide con la probabilidad de error de tipo I, es decir, de rechazar H0 cuando esta es cierta, se mueve en torno a α=0.05 para las tres pruebas (0.058, 0.054, 0.054). Las pruebas de Cramér-von Mises y Anderson Darling han funcionado ligeramente mejor que Kolmogorov-Smirnov, por tanto, ya que han rechazado menos.

En este caso vemos en los boxplots cómo los p-valores se distribuyen en un rango mucho mayor, al igual que en el histograma, donde vemos también que se distribuyen mediante una U(0,1). Esto se observa de nuevo en las gráficas que enfrentan la distribución empírica y la teórica de los p-valores.

Los resultados son de nuevo coherentes con la prueba que se está a realizar, ya que H0 sigue siendo la misma (que la distribución es una N(0,1)), y en este caso las muestras sí siguen la distribución especificada bajo la hipótesis nula. Entonces, que los p-valores sigan una U(0,1) indica que, en promedio, estamos tomando las decisiones correctas sobre si rechazar o no la hipótesis nula.

Parte B: Estimación no paramétrica de curvas

```
library(ggplot2)
library(sm)
```

Ejercicio 1

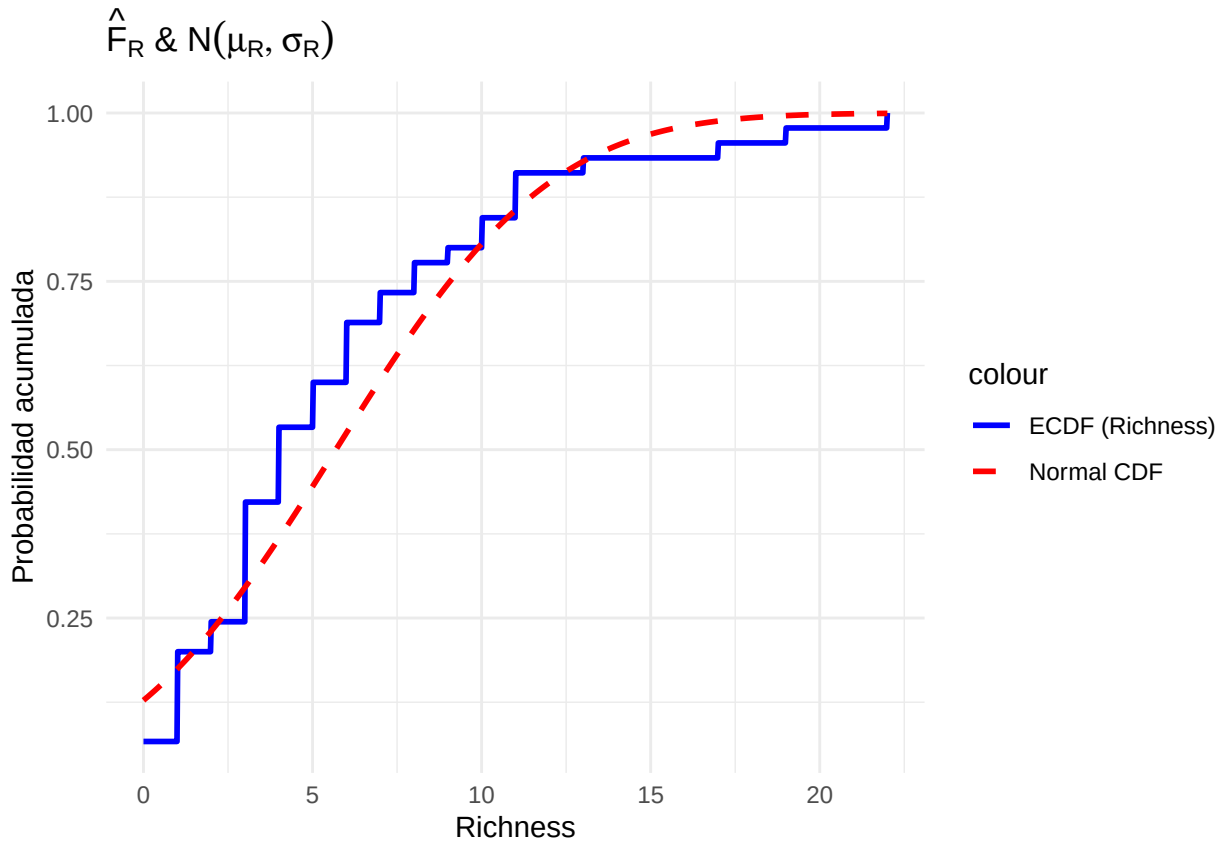
```
## Package 'sm', version 2.2-6.0: type help(sm) for summary information
```

```r
Richness <- RIKZ$Richness
NAP <- RIKZ$NAP
```

A continuación calculamos un estimador no paramétrico de la función de distribución de probabilidad $\hat{F}_R(.)$, la ECDF, y lo representamos en un único gráfico junto con la distribución de la normal con media y deviación típica estimadas con los registros de Richness.

```r
# Estimador no paramétrico de la función de distribución de probabilidad (ECDF)
F_R <- ecdf(Richness)

# Representar el estimador \hat{F}_R(.) y la distribución normal
mean_richness <- mean(Richness)
sd_richness <- sd(Richness)

# Crear un rango para las representaciones
grid <- seq(min(Richness), max(Richness), length.out = 1000)
normal_cdf <- pnorm(grid, mean = mean_richness, sd = sd_richness)

plot_df <- data.frame(
  x = grid,
  ECDF = F_R(grid),
  Normal_CDF = normal_cdf
)

ggplot(plot_df, aes(x = x)) +
  geom_line(aes(y = ECDF, color = "ECDF (Richness)"), size = 1) +
  geom_line(aes(y = Normal_CDF, color = "Normal CDF"), linetype = "dashed", size = 1) +
  labs(
    title = expression(paste(hat(F)[R], " & ", N(mu[R], sigma[R]))),
    x = "Richness",
    y = "Probabilidad acumulada"
  ) +
  scale_color_manual(values = c("ECDF (Richness)" = "blue", "Normal CDF" = "red")) +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Procedemos estimando P(Richness r) para r = 7 y r = 21.

```
p_7 <- F_R(7)
p_21 <- F_R(21)
cat("P(Richness <= 7):", p_7, "\n")
```

```
## P(Richness <= 7): 0.7333333
```

```
cat("P(Richness <= 21):", p_21, "\n")
```

```
## P(Richness <= 21): 0.9777778
```

Chequeamos la normalidad de Richness empleando una prueba basada en $\hat{F}_R(.)$, concretamente, el test de Shapiro Wilk.

```
shapiro.test(Richness)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Richness
## W = 0.86269, p-value = 7.9e-05
```

Como podemos observar, obtenemos un p-valor muy pequeño, por lo que rechazamos la hipósesis nula, concluyendo que la variable Richness no sigue una normal.

A continuación proporcionamos dos estimadores no paramétricos de la función de densidad de probabilidad de Richness ($\hat{f}_{h_1,R}(.)$ y $\hat{f}_{h_2,R}(.)$). Para ello, utilizamos dos parámetros de suavizado diferentes, h_1 y h_2, seleccionados por la "regla del dedo" y por el criterio plug-in de Sheater y Jones.

```
# Regla del dedo
h1 <- bw.nrd(Richness); h1
```

```
## [1] 1.847262
```

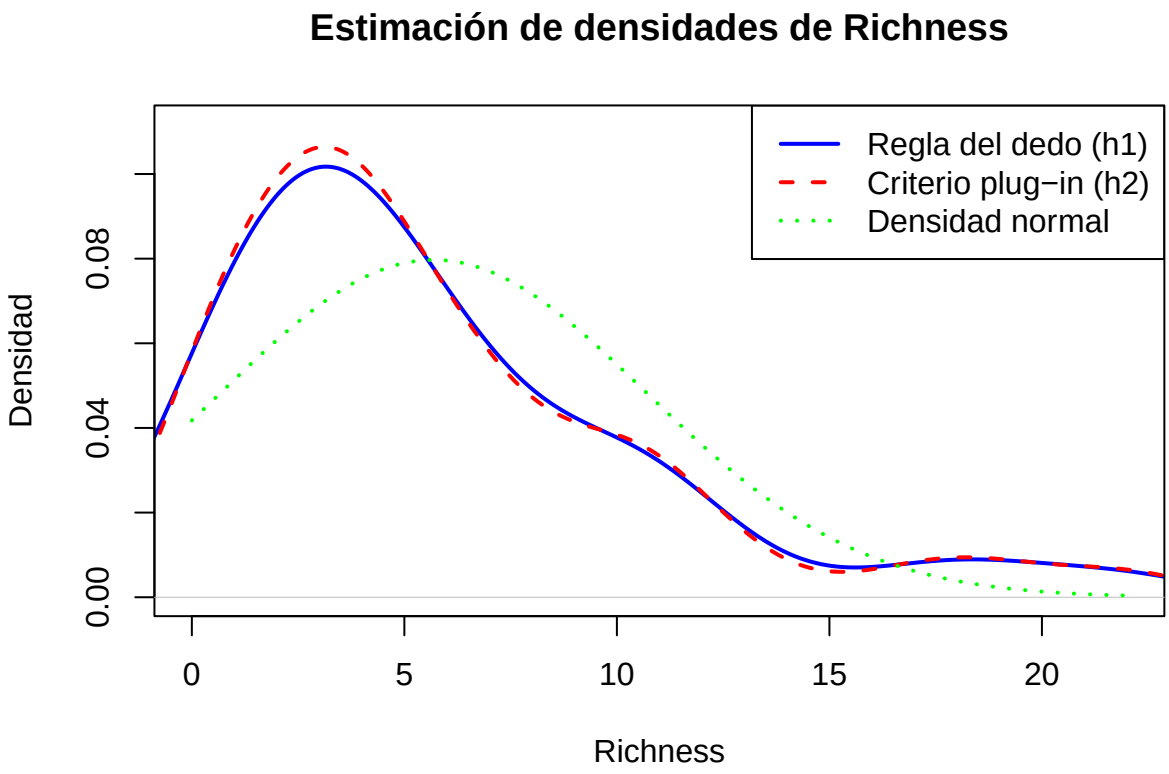
```
densidad_h1 <- density(Richness, bw = h1)

# Criterio plug-in de Sheater y Jones
h2 <- bw.SJ(Richness, method = "dpi"); h2
```

```
## [1] 1.599205
```

```
densidad_h2 <- density(Richness, bw = h2)
```

En la siguiente gráfica representamos $\hat{f}_{h_1,R}(.)$, $\hat{f}_{h_2,R}(.)$ y la densidad de la normal con media y desviación típica estimadas con los registros de Richness.

```
# Densidad normal basada en la media y desviación típica de Richness
densidad_normal <- function(x) {
  dnorm(x, mean = mean(Richness), sd = sd(Richness))
}

Y <- max(densidad_h1$y)
plot(densidad_h1, col = "blue", lwd = 2, main = "Estimación de densidades de Richness",
     xlab = "Richness", ylab = "Densidad", xlim = range(Richness), ylim=c(0,Y+0.01))
lines(densidad_h2, col = "red", lwd = 2, lty = 2)
curve(densidad_normal, col = "green", lwd = 2, lty = 3, add = TRUE)
legend("topright", legend = c("Regla del dedo (h1)", "Criterio plug-in (h2)", "Densidad normal"),
       col = c("blue", "red", "green"), lwd = 2, lty = c(1, 2, 3))
```



Estimamos el valor de la densidad de Richness en 7 y 21:

```
r_vals <- c(7, 21)
densidad_h1_vals <- approx(densidad_h1$x, densidad_h1$y, xout = r_vals)$y
densidad_h2_vals <- approx(densidad_h2$x, densidad_h2$y, xout = r_vals)$y

cat("Densidad en r = 7 y r = 21 (Regla del dedo):", densidad_h1_vals, "\n")
```

```
## Densidad en r = 7 y r = 21 (Regla del dedo): 0.05963407 0.007276637
```

```
cat("Densidad en r = 7 y r = 21 (Criterio plug-in):", densidad_h2_vals, "\n")
```

```
## Densidad en r = 7 y r = 21 (Criterio plug-in): 0.05806349 0.007338575
```

Chequeamos la normalidad de Richness empleando una prueba basada en $\hat{f}_R(.)$, concretamente evaluamos el ISE observado:

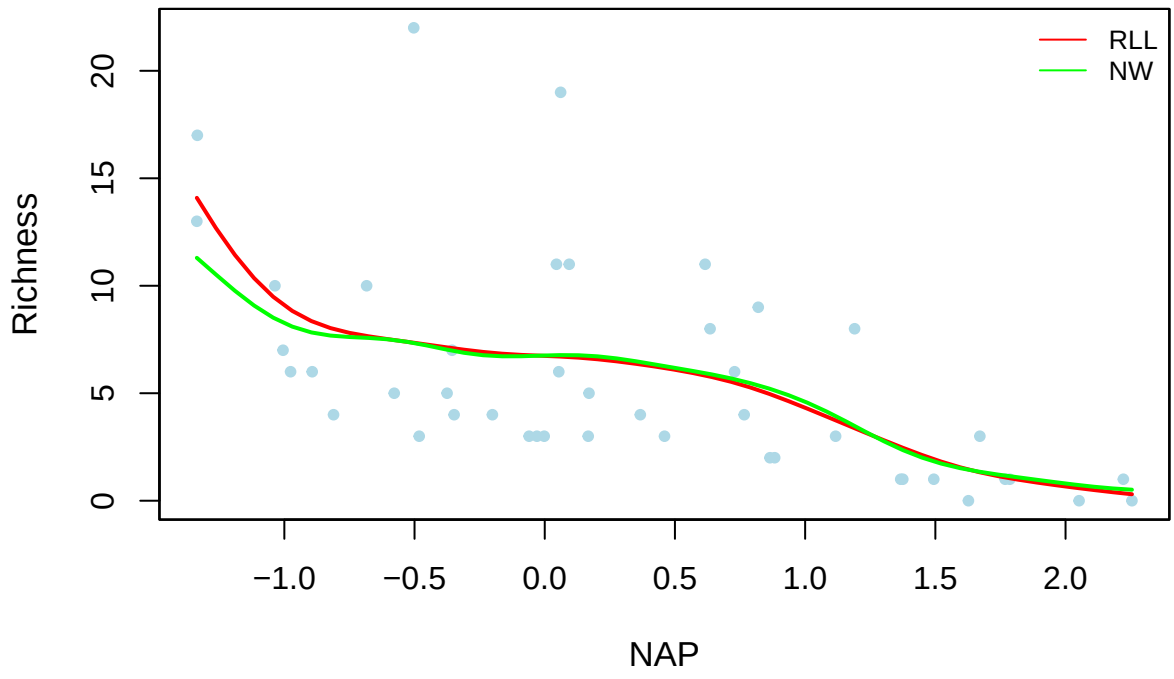
```
nise.observado <- nise(Richness); nise.observado
```

```
## [1] 0.02287104
```

En base al valor obtenido (0.02287104), rechazamos la normalidad de Richness a un nivel de significación del 5%.

A continuación modelizamos la relación entre Richness y NAP mediante un ajuste lineal local y con el estimador de Nadaraya-Watson.

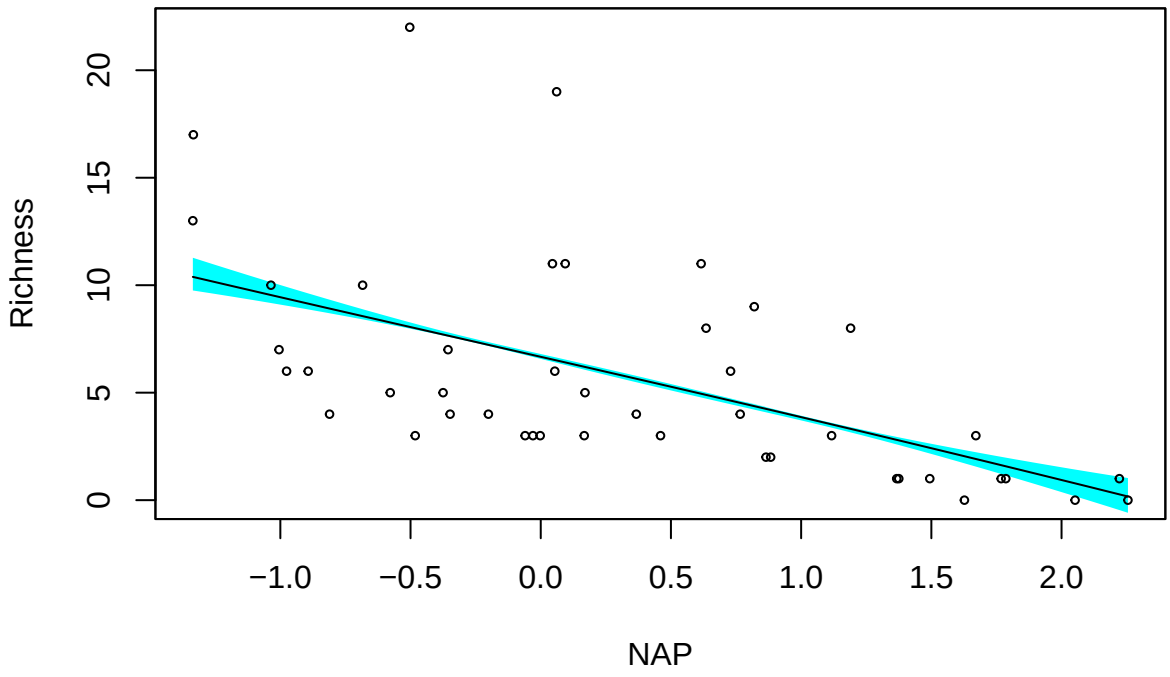
```r
# Crear un rango de valores para NAP para las predicciones
NAP_seq <- seq(min(NAP), max(NAP), length.out = 100)

plot(NAP, Richness, col = "lightblue", pch = 20, main = "",
     xlab = "NAP", ylab = "Richness", cex.lab = 1.1)

# Ajuste lineal local
sm.regression(NAP, Richness, poly.index = 1, add=TRUE, col="red", lwd=2)  # Ajuste RLL

# Estimador de Nadaraya-Watson
sm.regression(NAP, Richness, poly.index = 0, add=TRUE, col="green", lwd=2)  # Ajuste NW

legend("topright", legend = c("RLL", "NW"), col = c("red", "green"),
       lty = 1, cex = 0.8, bty = "n")
```



Aunque ambos ajustes son similares y parece que modelan correctamente los datos, vemos que el ajuste de Nadaraya-Watson sufre efecto frontera en los puntos x cercanos al límite inferior del dominio de X, donde vemos que realiza una peor estimación que el ajuste local lineal, el cual lo combate de forma natural.

Chequeamos la linealidad de la relación:

```r
ajuste_lineal <- sm.regression(NAP, Richness, method="aicc", model = "linear")
```

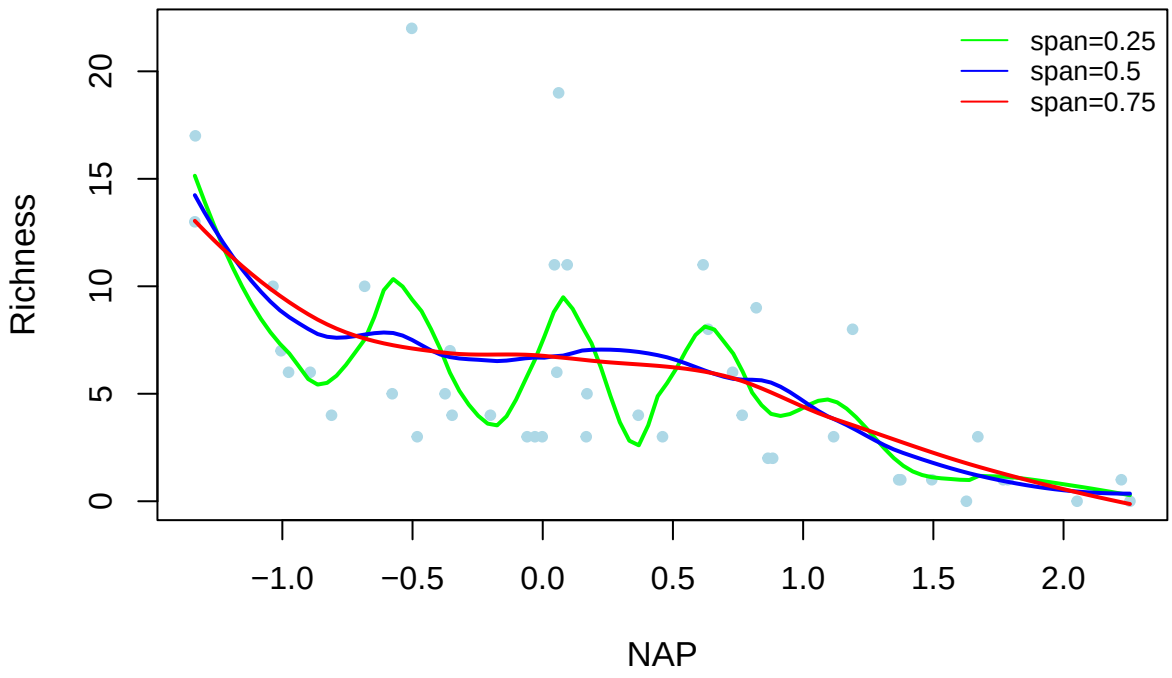



```
## Test of linear model:  significance =  0.855
```

En base al p-valor obtenido, podemos asumir linealidad entre Richness y NAP.

Ahora estimamos la relación entre Richness y NAP con el algoritmo loess.

```r
loess_fit_1 <- loess(Richness ~ NAP, span = 0.25)
loess_fit_2 <- loess(Richness ~ NAP, span = 0.5)
loess_fit_3 <- loess(Richness ~ NAP, span = 0.75)

# Predecir valores de Richness usando el modelo loess
Richness_loess_1 <- predict(loess_fit_1, newdata = data.frame(NAP = NAP_seq))
Richness_loess_2 <- predict(loess_fit_2, newdata = data.frame(NAP = NAP_seq))
Richness_loess_3 <- predict(loess_fit_3, newdata = data.frame(NAP = NAP_seq))

plot(NAP, Richness, col = "lightblue", pch = 20, main = "",
     xlab = "NAP", ylab = "Richness", cex.lab = 1.1)
lines(NAP_seq, Richness_loess_1, col = "green", lwd = 2)
lines(NAP_seq, Richness_loess_2, col = "blue", lwd = 2)
lines(NAP_seq, Richness_loess_3, col = "red", lwd = 2)
legend("topright", legend = c("span=0.25","span=0.5","span=0.75"), col = c("green","blue","red"),
       lty = 1, cex = 0.8, bty = "n")
```



Tras probar con diferentes valores, parece que un valor de span adecuado es 0.75, ya que con valores inferiores se produce un sobreajuste, mientras que con este valor el sesgo no parece demasiado elevado. Este valor significa que entran el 75% de los datos muestrales más próximos a cada punto en el entorno donde se realiza el ajuste polinómico local en cada uno de dichos puntos.

A continuación mostramos los grados de libertad y la varianza residual del ajuste:

```
cat("Grados de libertad del ajuste: ", loess_fit_3$trace.hat, "\n")
```

```
## Grados de libertad del ajuste:  5.213602
```

```
cat("Varianza residual del ajuste: ", (loess_fit_3$s)^2)
```

```
## Varianza residual del ajuste:  17.50659
```