TAREA MAS

María Pallares Diez

2025-05-29

INTRO

En Nueva York, los espectaculares puentes que atraviesan el East River cuentan con carriles bici-peatonales que permiten cruzar a sus "Boros" vecinos sin necesidad de recurrir al automóvil o al transporte público, a menudo saturado. En este ejercicio nos centraremos en el Puente de Manhattan, una arteria ciclista clave que conecta el corazón de Manhattan con Brooklyn.

Usaremos el archivo biciesM, que recoge datos diarios sobre el número de bicis por minuto (nmin) que atravesaron el puente de Manhattan (Nueva York) en el verano de 2017. Otras variables relevantes son: tmean: temperatura en grados Farenheit, lprec: logaritmo del volumen de precipitación en los días de lluvia y algunas variables de calendario. Nos proponemos analizar cómo estos factores influyen en el tránsito de bicicletas a través del Puente de Manhattan durante estos meses (de junio a septiembre), que también son los mayor afluencia turística.

EJERCICIO

Responde razonadamente a las siguientes preguntas, partiendo del siguiente modelo tentativo:

```
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```
load("biciesM.RData")
biciesM$precipitation<-ifelse(biciesM$lprec==0,0, exp(biciesM$lprec))
m1<-gam(nmin~s(tmean)+s(day,bs="bs")+s(precipitation)+month+dow,
data=biciesM, family=poisson(link="log"), na.action=na.exclude)
summary(m1)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## nmin ~ s(tmean) + s(day, bs = "bs") + s(precipitation) + month +
##     dow
##
```

```
## Parametric coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.70381    0.36333   7.442 9.93e-14 ***
## month         0.01170    0.05145   0.227    0.820
## dowjueves     0.36363    0.05619   6.471 9.73e-11 ***
## dowlunes      0.32226    0.05760   5.595 2.21e-08 ***
## dowmartes     0.41788    0.05658   7.386 1.51e-13 ***
## dowmiercoles  0.41460    0.05585   7.424 1.14e-13 ***
## dowsabado     0.04967    0.06246   0.795    0.426
## dowviernes    0.27345    0.05847   4.677 2.91e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                     edf Ref.df  Chi.sq p-value
## s(tmean)          2.810  3.555  47.385  <2e-16 ***
## s(day)            1.000  1.000   0.014   0.906
## s(precipitation) 5.166  6.087 177.150  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.793   Deviance explained = 79.5%
## UBRE = -0.2919  Scale est. = 1         n = 214
```

1. Explica los suavizadores en el modelo. ¿Qué habría que cambiar para ajustar un natural spline sin penalizar de día, con nodos en sus cuartiles?

En el modelo se usan suavizadores `s()` que, por defecto, son splines penalizados (típicamente thin plate regression splines) para modelar relaciones no lineales entre la variable dependiente (`nmin`, número de bicicletas por minuto) y las variables independientes como la temperatura (`tmean`), el día (`day`) y la precipitación (`precipitation`). El término `s(day, bs="bs")` indica el uso de un B-spline penalizado para la variable día.

Para ajustar un natural spline *sin penalizar* para la variable día y situar los nodos en sus cuartiles, habría que sustituir el suavizador penalizado por un spline natural no penalizado, lo que se consigue utilizando la función `ns()` de la librería `splines` y especificando los nodos (knots) en los cuartiles de la variable `day`.

2. Realiza las acciones que consideres oportunas y formula el modelo que hayas elegido.

```r
library(splines)
# Calcular los cuartiles para day (excluyendo posibles NA)
knots_day <- quantile(biciesM$day, probs = c(0.25, 0.5, 0.75), na.rm = TRUE)

# Ajuste con natural spline sin penalizar
m_ns <- gam(nmin ~ s(tmean) + ns(day, knots = knots_day) + s(precipitation) + month + dow,data = biciesl

summary(m_ns)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
```

```
## nmin ~ s(tmean) + ns(day, knots = knots_day) + s(precipitation) +
##     month + dow
##
## Parametric coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    2.70651    0.20677  13.090  < 2e-16 ***
## ns(day, knots = knots_day)1 -0.01268    0.19931  -0.064    0.949
## ns(day, knots = knots_day)2  0.02119    0.25864   0.082    0.935
## ns(day, knots = knots_day)3  0.01550    0.42622   0.036    0.971
## ns(day, knots = knots_day)4 -0.05739    0.30910  -0.186    0.853
## month                          0.01040    0.05180   0.201    0.841
## dowjueves                      0.36292    0.05622   6.456 1.08e-10 ***
## dowlunes                       0.32235    0.05763   5.593 2.23e-08 ***
## dowmartes                      0.41776    0.05661   7.380 1.58e-13 ***
## dowmiercoles                   0.41470    0.05585   7.425 1.13e-13 ***
## dowsabado                      0.04979    0.06247   0.797    0.425
## dowviernes                     0.27235    0.05851   4.655 3.24e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                     edf Ref.df Chi.sq p-value
## s(tmean)          2.758  3.508  28.91 7.8e-06 ***
## s(precipitation) 5.167  6.087 174.24 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.791   Deviance explained = 79.5%
## UBRE = -0.265  Scale est. = 1         n = 214
```

3. Da una medida de bondad de ajuste. Indica e interpreta sus grados de libertad efectivos.

```r
# Extraer resumen de ambos modelos
summary_m1   <- summary(m1)
summary_mns  <- summary(m_ns)

# Crear tabla de comparación
tabla_comp <- data.frame(
  Modelo = c("m1: s(day, bs='bs')", "m_ns: ns(day, knots=cuartiles)"),
  Deviance_exp = c(
    round(summary_m1$dev.expl * 100, 1),
    round(summary_mns$dev.expl * 100, 1)
  ),
  R2_adj = c(
    round(summary_m1$r.sq, 3),
    round(summary_mns$r.sq, 3)
  ),
  edf_day = c(
    summary_m1$s.table["s(day)", "edf"],
    NA # No hay edf para ns(), ponemos NA
  ),
  pval_day = c(
```

```
    summary_m1$s.table["s(day)", "p-value"],
    # Para ns(), extraemos el p-valor global (ANOVA), si lo deseas:
    anova(m_ns)$"Pr(>F)"[2]
  ),
  edf_tmean = c(
    summary_m1$s.table["s(tmean)", "edf"],
    summary_mns$s.table["s(tmean)", "edf"]
  ),
  pval_tmean = c(
    summary_m1$s.table["s(tmean)", "p-value"],
    summary_mns$s.table["s(tmean)", "p-value"]
  ),
  edf_precip = c(
    summary_m1$s.table["s(precipitation)", "edf"],
    summary_mns$s.table["s(precipitation)", "edf"]
  ),
  pval_precip = c(
    summary_m1$s.table["s(precipitation)", "p-value"],
    summary_mns$s.table["s(precipitation)", "p-value"]
  )
)

tabla_comp
```

```
##                           Modelo Deviance_exp R2_adj  edf_day  pval_day
## 1            m1: s(day, bs='bs')         79.5  0.793 1.000042 0.9063121
## 2 m_ns: ns(day, knots=cuartiles)         79.5  0.791       NA 0.9063121
##   edf_tmean   pval_tmean edf_precip pval_precip
## 1  2.810402 0.000000e+00   5.166409           0
## 2  2.758315 7.799794e-06   5.166817           0
```

Para comparar los modelos ajustados, analizamos la **deviance explicada**, el **R-cuadrado ajustado** y los **grados de libertad efectivos** (edf) asociados a los suavizadores.

Medidas de ajuste

- El modelo **m1** (`s(day, bs = "bs")`) presenta una **deviance explicada del 79.5 %** y un **R^2 ajustado de 0.793**.
- El modelo **m__ns** (`ns(day, knots = nodos_day)`) muestra valores casi idénticos: **deviance explicada del 79.5 %** y **R^2 ajustado de 0.791**.

Interpretación:
Ambos modelos logran explicar prácticamente la misma proporción de la variabilidad en el número de bicicletas, por lo que ambos ajustan de manera similar a los datos.

Grados de libertad efectivos (edf)

- Para **m1**, el suavizador de **día** (`s(day, bs = "bs")`) tiene un **edf de 1**, lo que indica que el modelo ha estimado que la relación entre el día y el tránsito de bicicletas es esencialmente **lineal** (no ha encontrado evidencia de una tendencia no lineal relevante).

- Para **m_ns**, al utilizar un spline natural sin penalizar, el ajuste de día se realiza mediante una base de natural splines con los nodos en los cuartiles. En este caso, no se calcula un "edf" propiamente dicho, ya que los parámetros del spline quedan fijos y corresponden a los grados de libertad definidos por el número de nodos. Sin embargo, los coeficientes asociados a los términos del spline (`ns(day, knots = ...)`) tampoco resultan significativos, lo que indica que, igualmente, **no se observa una tendencia temporal compleja**.

Comparación de suavizadores

- Los suavizadores para **temperatura** y **precipitación** en ambos modelos presentan edf > 1 y resultan significativos ($p < 0.001$), lo que confirma relaciones **no lineales** entre estas variables y el tránsito de bicicletas.
- En cambio, el término de **día** no es significativo en ninguno de los dos modelos ($p > 0.9$), y tanto el modelo penalizado (`s(day)`) como el natural spline (`ns(day)`) apuntan a que **no hay patrones temporales apreciables** dentro del rango de días considerados.

Ambos modelos son equivalentes en ajuste y complejidad, y ambos sugieren que el efecto del día no es relevante ni lineal ni no lineal, mientras que temperatura y precipitación muestran relaciones no lineales y estadísticamente significativas con el tránsito ciclista.

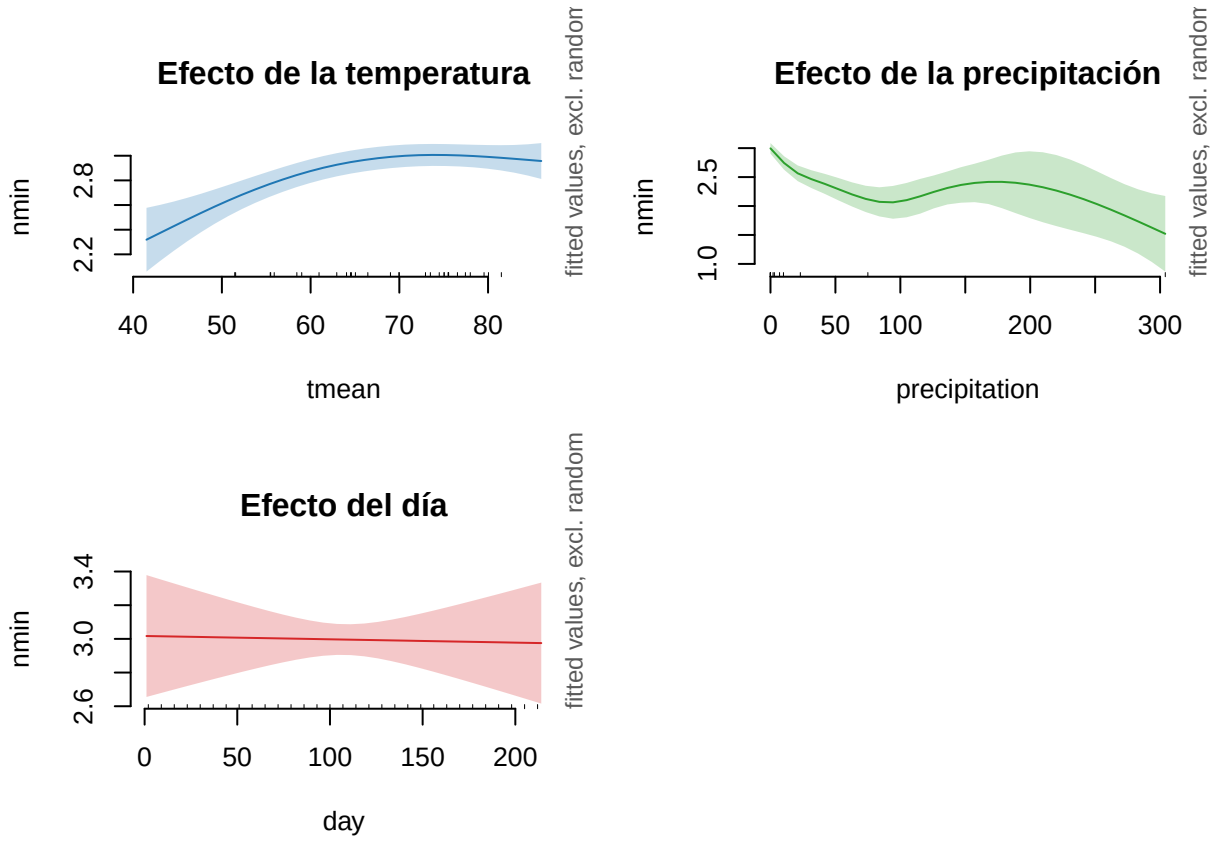
Ambos modelos ofrecen valores muy similares en cuanto a deviance explicada y R^2 ajustado, situándose en torno al 79.5 %. Sin embargo, el modelo `m1` (que utiliza el suavizador penalizado para `day`) presenta un **R^2 ajustado ligeramente superior** y el mismo porcentaje de deviance explicada, lo que indica un ajuste marginalmente mejor a los datos. Además, el término de día no resulta significativo en ninguno de los dos modelos y ambos capturan la no linealidad relevante en las variables de temperatura y precipitación. Por tanto, **me quedo con el modelo `m1`**, ya que logra el mejor ajuste posible para estos datos, sin aumentar innecesariamente la complejidad del modelo. No obstante, si el criterio principal fuera cumplir el enunciado estrictamente (natural spline sin penalizar), se podría justificar también la elección de `m_ns`, aunque con un ajuste prácticamente idéntico.

4. Representa e interpreta los efectos y las relaciones no lineales en el modelo elegido.

A continuación se muestran los gráficos de los efectos suavizados estimados por el modelo GAM para las tres variables principales: temperatura media, precipitación y día. Cada gráfico representa la relación ajustada por el modelo entre la variable correspondiente y el número medio de bicicletas por minuto, junto con su intervalo de confianza al 95 %.

```r
# Librerías
library(itsadug)
library(mgcv)

# Disposición de la ventana gráfica
par(mfrow = c(2, 2))  # 1 fila, 3 columnas

# Gráfico 1: Efecto de la temperatura
plot_smooth(m1, view = "tmean", rug = TRUE, shade = TRUE,
            col = "#1F77B4", se.col = "lightblue",
            main = "Efecto de la temperatura",
            print.summary = FALSE)

# Gráfico 2: Efecto de la precipitación
plot_smooth(m1, view = "precipitation", rug = TRUE, shade = TRUE,
```

```
            col = "#2CA02C", se.col = "lightgreen",
            main = "Efecto de la precipitación",
            print.summary = FALSE)

# Gráfico 3: Efecto del día
plot_smooth(m1, view = "day", rug = TRUE, shade = TRUE,
            col = "#D62728", se.col = "salmon",
            main = "Efecto del día",
            print.summary = FALSE)
```



Efecto de la temperatura (tmean):

- Se observa una **relación no lineal**. El tránsito de bicicletas aumenta al incrementarse la temperatura, alcanzando un máximo alrededor de los 70 °F, para después estabilizarse o incluso disminuir levemente. Esto sugiere que hay un rango óptimo de temperatura para el uso de la bicicleta: temperaturas bajas o excesivamente altas son menos favorables para el tránsito ciclista.

Efecto de la precipitación:

- La relación estimada también es **no lineal**. A medida que la precipitación aumenta, el número de bicicletas disminuye notablemente al principio (lo que resulta intuitivo: la lluvia reduce el uso de la bicicleta). Sin embargo, para valores intermedios-altos la relación se suaviza y puede aparecer cierta estabilización o incluso un pequeño repunte, aunque con mayor incertidumbre, posiblemente por la escasez de días con precipitaciones extremas en la muestra.

Efecto del día:

- El efecto del día sobre el tránsito de bicicletas es **prácticamente lineal y no significativo**. La banda de confianza es muy ancha y la función suavizada es prácticamente plana, indicando que no existe una tendencia temporal clara en el periodo analizado (junio-septiembre). Esto es coherente con los resultados numéricos, donde el grado de libertad efectivo para este suavizador es cercano a 1 y el p-valor no es significativo.

Conclusión:
El modelo detecta e interpreta correctamente relaciones no lineales importantes para la temperatura y la precipitación, mientras que el efecto del día no aporta información relevante sobre el tránsito de bicicletas en el Puente de Manhattan durante el verano de 2017.

5. Proporciona una predicción con IC(95 %) para el tránsito de bicis el 4 de julio (fila 96 del archivo).

```
nueva_obs <- biciesM[96, ]
print(nueva_obs)
```

```
##          date month day       dow tmean   tmeanc lprec lprec_2c nmin
## 96 2017-07-05     7  96 miercoles    78 9.885514     0     1.no   28
##    precipitation
## 96             0
```

```
# Predicción puntual y error estándar en la escala del predictor lineal (log)
pred <- predict(m1, newdata = nueva_obs, type = "link", se.fit = TRUE)

# Intervalo de confianza al 95 % en la escala log
ic_inf_log <- pred$fit - 1.96 * pred$se.fit
ic_sup_log <- pred$fit + 1.96 * pred$se.fit

# Escala original
prediccion <- exp(pred$fit)
ic_inf <- exp(ic_inf_log)
ic_sup <- exp(ic_sup_log)

# Mostrar resultado incluyendo la fecha
data.frame(
  Fecha = nueva_obs$date,
  Prediccion_n_bicis_min = round(prediccion, 2),
  IC_inf_95 = round(ic_inf, 2),
  IC_sup_95 = round(ic_sup, 2)
)
```

```
##         Fecha Prediccion_n_bicis_min IC_inf_95 IC_sup_95
## 96 2017-07-05                 30.43     27.89     33.21
```

El modelo predice que el **5 de julio de 2017** (fila 96), el número esperado de bicicletas por minuto que cruzan el Puente de Manhattan es de **30.43**, con un intervalo de confianza al 95 % entre **27.89** y **33.21** bicicletas por minuto.

Esto significa que, bajo las condiciones meteorológicas y de calendario de ese día, el modelo estima que el tránsito real se encontrará, con un 95 % de confianza, entre 27.89 y 33.21 bicicletas por minuto.

```
nueva_obs <- biciesM[95, ]
print(nueva_obs)
```

```
##          date month day    dow tmean   tmeanc lprec lprec_2c nmin precipitation
## 95 2017-07-04     7  95 martes 76.45 8.335514     0     1.no   17             0
```

```
# Predicción puntual y error estándar en la escala del predictor lineal (log)
pred <- predict(m1, newdata = nueva_obs, type = "link", se.fit = TRUE)

# Intervalo de confianza al 95 % en la escala log
ic_inf_log <- pred$fit - 1.96 * pred$se.fit
ic_sup_log <- pred$fit + 1.96 * pred$se.fit

# Escala original
prediccion <- exp(pred$fit)
ic_inf <- exp(ic_inf_log)
ic_sup <- exp(ic_sup_log)

# Mostrar resultado incluyendo la fecha
data.frame(
  Fecha = nueva_obs$date,
  Prediccion_n_bicis_min = round(prediccion, 2),
  IC_inf_95 = round(ic_inf, 2),
  IC_sup_95 = round(ic_sup, 2)
)
```

```
##         Fecha Prediccion_n_bicis_min IC_inf_95 IC_sup_95
## 95 2017-07-04                 30.68     28.08     33.52
```

Para el **4 de julio de 2017** (fila 95), el modelo predice que el número esperado de bicicletas por minuto que cruzan el Puente de Manhattan es de **30.68**, con un intervalo de confianza al 95 % entre **28.08** y **33.52** bicicletas por minuto.

Esto significa que, considerando las condiciones meteorológicas y de calendario de ese día, el modelo estima con un 95 % de confianza que el tránsito real de bicicletas estará comprendido entre 28.08 y 33.52 bicicletas por minuto.

6. ¿Cuál sería efecto (con IC(95 %)) sobre el tránsito de bicis de un aumento de 10 žF en la temperatura desde el valor asociado con el máximo transito.

```
# 1. Buscar la temperatura con máximo tránsito predicho
t_seq <- seq(min(biciesM$tmean, na.rm = TRUE), max(biciesM$tmean, na.rm = TRUE), length = 100)

# Creamos una observación tipo (puedes ajustar month, dow, etc. a valores frecuentes o al 4 de julio)
nueva_obs <- biciesM[95, ]  # 4 de julio

# Generamos un dataframe variando tmean, el resto se mantiene igual
df_pred <- nueva_obs[rep(1, 100), ]
```

```r
df_pred$tmean <- t_seq

# Predecimos el tránsito esperado para cada tmean
preds <- predict(m1, newdata = df_pred, type = "response")
t_optimo <- t_seq[which.max(preds)]  # Temperatura con máximo tránsito

# Creamos dos nuevas observaciones: t_optimo y t_optimo + 10
obs1 <- nueva_obs
obs2 <- nueva_obs
obs1$tmean <- t_optimo
obs2$tmean <- t_optimo + 10

# Predicción y error estándar en la escala log
pred1 <- predict(m1, newdata = obs1, type = "link", se.fit = TRUE)
pred2 <- predict(m1, newdata = obs2, type = "link", se.fit = TRUE)

# Efecto multiplicativo (escala logarítmica)
diferencia_log <- pred2$fit - pred1$fit
se_dif <- sqrt(pred1$se.fit^2 + pred2$se.fit^2)

# IC 95% para el efecto multiplicativo
efecto <- exp(diferencia_log)
ic_inf <- exp(diferencia_log - 1.96 * se_dif)
ic_sup <- exp(diferencia_log + 1.96 * se_dif)

# Mostrar resultado
data.frame(
  T_optimo = t_optimo,
  T_optimo_mas10 = t_optimo + 10,
  Efecto_multiplicativo = round(efecto, 3),
  IC_inf_95 = round(ic_inf, 3),
  IC_sup_95 = round(ic_sup, 3)
)
```

```
##    T_optimo T_optimo_mas10 Efecto_multiplicativo IC_inf_95 IC_sup_95
## 95 73.86364       83.86364                 0.964     0.834     1.114
```

Al analizar el modelo, se observa que la temperatura asociada al máximo tránsito de bicicletas es de **73.86 °F**. Si aumentamos la temperatura en 10 °F hasta **83.86 °F**, el **efecto multiplicativo** estimado sobre el tránsito de bicicletas es de **0.964**, con un intervalo de confianza al 95 % entre **0.834** y **1.114**.

Interpretación:
Esto significa que, al incrementar la temperatura 10 °F por encima del valor óptimo, el tránsito de bicicletas tiende a disminuir ligeramente (un 3.6 % menos), aunque el intervalo de confianza incluye el valor 1. Por tanto, este efecto **no es estadísticamente significativo**: no se puede afirmar con suficiente evidencia que un aumento adicional de temperatura a partir del máximo suponga una reducción real del tránsito ciclista.