

Tarea 5

Luciano Andres Juárez López

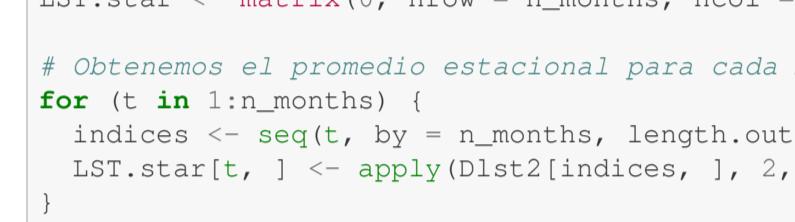
Ejercicio 1

```
In [1]: # install.packages("fda")  
  
In [1]: # Leemos los datos de NDVI  
load("../LSTMNDVI.RData")  
  
In [1]: head(Dlst2)  
print(dim(Dlst2))  
  
A data.frame: 6 x 258  
   dataos.2  dataos.4  dataos.6  dataos.7  dataos.8  dataos.9  dataos.10  dataos.11  dataos.13 ...  dataos.201  dataos.202  dataos.203  dataos.204  dataos.205  
   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> ...  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  
1 21.27639 23.07269 23.27639 23.81138 23.22135 23.87339 23.87339 23.87339 23.87339 ... 21.10218 20.98535 22.01137 26.47679 23.0777  
2 25.13350 23.98051 36.10605 25.38196 24.27639 33.89231 27.81846 29.15033 21.74249 ... 27.43577 28.04759 26.95725 36.3344  
3 38.99210 38.89981 40.60540 29.71217 33.29972 37.91205 29.88677 38.73722 ... 38.57750 45.71876 40.70452 40.07903 48.18657 39.406  
4 35.05453 34.98200 36.26680 27.63031 34.25014 33.50952 38.40705 41.61713 39.35782 42.41011 35.34961 ... 40.56469 43.19626 44.36562 50.94237 42.0717  
5 43.63716 33.42154 34.70548 34.21235 33.67336 38.40705 41.61713 39.35782 42.41011 35.34961 ... 40.56469 43.19626 44.36562 50.94237 42.0717  
6 44.04043 33.42154 34.70548 34.21235 33.67336 38.40705 41.61713 39.35782 42.41011 35.34961 ... 40.56469 43.19626 44.36562 50.94237 42.0717  
[1] 72 158
```

```
In [1]: head(Dndv12)  
print(dim(Dndv12))  
  
A data.frame: 6 x 158  
   dataos.2  dataos.3  dataos.4  dataos.6  dataos.7  dataos.8  dataos.9  dataos.10  dataos.11  dataos.13 ...  dataos.201  dataos.202  dataos.203  dataos.204  dataos.205  
   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> ...  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  
1 0.0681570 0.0995841 0.0682504 0.0600433 0.0707605 0.3440705 0.3819568 0.6159417 0.3186409 0.07981670 ... 0.1331171 0.0870167 0.1532802 0.0  
2 0.1341916 0.2513807 0.0666700 0.0576084 0.0864700 0.3621270 0.2918360 0.4465471 0.6891246 0.3890856 0.08425577 0.2453105 0.06658348 0.1744812 0.0  
3 0.0839544 0.1028100 0.0975084 0.0912847 0.0742700 0.2918360 0.4465471 0.6891246 0.3890856 0.08425577 0.2453105 0.06658348 0.1744812 0.0  
4 0.07615615 0.1258200 0.0853423 0.0794154 0.0672793 0.2918360 0.4465471 0.6891246 0.3890856 0.08425577 0.2453105 0.06658348 0.1744812 0.0  
5 0.1035804 0.1398100 0.09485308 0.0818342 0.0794154 0.2918360 0.4465471 0.6891246 0.3890856 0.08425577 0.2453105 0.06658348 0.1744812 0.0  
6 0.10205212 0.139852190 0.07474834 0.0380260 0.07303713 0.3870984 0.4151791 0.5620583 0.4060087 0.1810697 ... 0.2264134 0.1001990 0.1765547 0.0  
[1] 72 158
```

Respuesta escalar $\max_{t \in T} \text{LST}(t)$ y predictor funcional NDVI(t).

```
In [1]: # Definimos la respuesta escalar  
Lst.max <- apply(Dlst2, 2, max)  
  
In [1]: # Buscamos el numero de base, con el cual no haya un sobreajuste  
nbasis.v <- c(10, 15, 20, 25, 30, 35, 40, 50, 60)  
  
# Suponemos y graficar para cada valor de nbasis  
par(mfrow = c(2, 2)) # Configuración para múltiples gráficos  
for (nbasis in nbasis.v) {  
  NVDI.basis <- create.fourier.basis(rangeval = c(0, 72), nbasis = nbasis)  
  NDVI.fd <- smooth.basis(1:t, Dndv12, NVDI.basis)$fd  
  # Graficar las curvas  
  plot(NDVI.fd, main = paste("nbasis =", nbasis), xlab = "Tiempo (t)", ylab = "NDVI (t)")  
}
```



De acuerdo a los graficos, podemos observar que la mejor base es cuando nbasis es igual a 15, pues al comparar con valores más altos se observa que las curvas se vuelven más complejas e irregulares, lo que sugiere un posible sobreajuste al ruido de los datos.

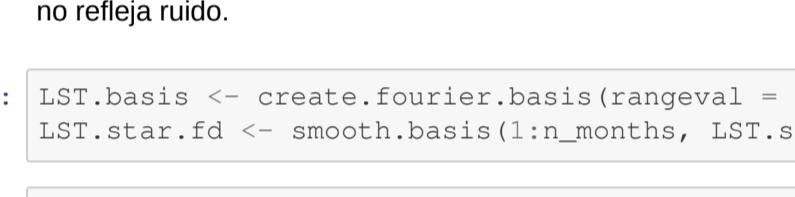
```
In [1]: # Definimos el predictor funcional  
NDVI.basis <- create.fourier.basis(rangeval = c(0, 72), nbasis = 15)  
NDVI.fd <- smooth.basis(1:t, Dndv12, NVDI.basis)$fd  
  
model.escal <- fRegress(Lst.max - NDVI.fd, method = "FRegres")
```

```
In [1]: # Cuando la respuesta es escalar, la predicción se encuentra en:  
Lst.predict <- model.escal$yhatfdobj
```

Datos Vg a ajuste
plot(Lst.max, Lst.predict, type = "p", pch = 2, main = "Observados vs Predichos", xlab = "Observado", ylab = "Predicho")

lines(Lst.predict, Lst.max, lty = 2)

Observados vs Predichos



La mayoría de los puntos parecen alinearse bien con la linea roja. Esto sugiere que el modelo capta adecuadamente la relación entre NDVI y Lstmax.

```
In [1]: # RMSE  
rmse <- sqrt(mean((Lst.max - Lst.predict)^2))  
print(paste("RMSE del modelo:", round(rmse, 3)))
```

[1] "RMSE del modelo: 2.553"

El RMSE igual a 2.553, nos sugiere un modelo razonablemente preciso

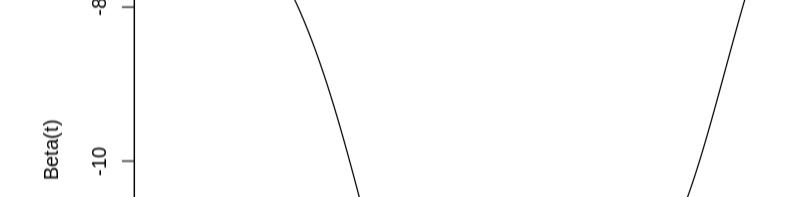
```
In [1]: names(model.escal$betaestlist)
```

"const" "NDVI.fd"

```
In [1]: # Coeficiente funcional de regresión estimado  
plot(model.escal$betaestlist[[2]], main = "Coeficiente funcional asociado a NDVI(t)", xlab = "Tiempo (t)", ylab = "Beta(t)")
```

'done'

Coeficiente funcional asociado a NDVI(t)



La gráfica muestra que el efecto del NDVI(t) en la temperatura media superficial cambia a lo largo del tiempo. Cuando los valores de beta son positivos, la vegetación está relacionada con un aumento de la temperatura. En cambio, cuando los valores son negativos, la vegetación ayuda a reducir la temperatura, mostrando un efecto de enfriamiento.

Respuesta funcional $LST^*(t) = \frac{1}{6} \sum_{k=0}^5 LST(t+12k)$ con $t = 1, \dots, 12$ y predictor escalar STID(NDVI).

```
In [1]: # Definimos la respuesta funcional  
n_months <- 12 # Número de meses  
n_sites <- ncol(Dlst2) # Número de sitios  
Lst.star <- matrix(0, nrow = n_months, ncol = n_sites)  
  
# Obtenemos el promedio estacional para cada mes  
for (t in 1:n_months) {  
  indices <- seq(1, by = n_months, length.out = 6)  
  Lst.star[t, ] <- apply(Dlst2[indices, ], 2, sum)/6  
}
```

Buscamos el numero de base, con el cual no haya un sobreajuste

nbasis.v <- c(10, 15, 20, 25, 30, 35, 40, 50, 60)

Suponemos y graficar para cada valor de nbasis

par(mfrow = c(2, 2)) # Configuración para múltiples gráficos

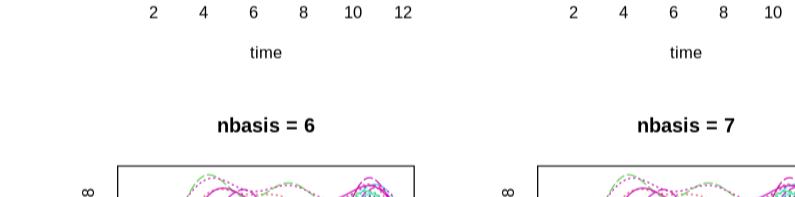
for (nbasis in nbasis.v) {
 Lst23.basis <- create.fourier.basis(rangeval = c(1, 12), nbasis = nbasis)
 Lst23.fd <- smooth.basis(1:t, Lst.star, Lst23.basis)\$fd

Graficar LST^*(t) para todos los sitios

plot(Lst23.fd, main = paste("nbasis =", nbasis))

'done'

Curvas observadas vs predichas



De acuerdo a estas graficas, seleccionamos nbasis igual a 6, pues captura adecuadamente las tendencias estacionales a lo largo de los meses, de igual forma no refleja ruido.

```
In [1]: Lst23.basis <- create.fourier.basis(rangeval = c(1, 12), nbasis = 6)  
Lst.star.fd <- smooth.basis(1:t, Lst.star, Lst23.basis)$fd
```

```
In [1]: source("../STD.R")
```

Calculamos STID para NDVI

STID.NVDI <- STDfDts(NVDI, data.frame(t(Dndv12)))

```
In [1]: # Ajustamos el modelo funcional con predictor escalar STID(NDVI)
```

model.funcional <- fRegress(Lst.star.fd - STID.NVDI)

```
In [1]: Lst.star.pred <- model.funcional$yhatfdobj
```

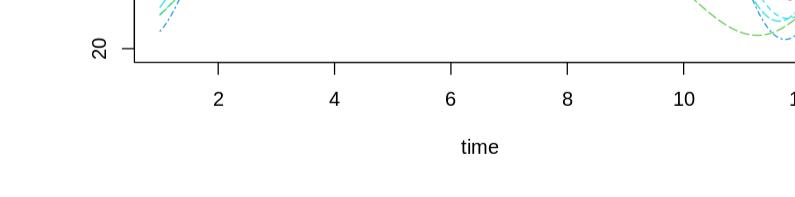
plot(Lst23.fd, main = "Curvas observadas vs predichas")

lines(Lst.star.pred, Lst23.fd, col = "red", lty = 2)

legend("topright", legend = c("Observado", "Predicho"), col = c("black", "red"), lty = c(1, 2))

'done'

Curvas observadas vs predichas



Podemos observar que las predicciones son suaves y no presentan oscilaciones extrañas. Por lo que podemos decir que el modelo está capturando correctamente los patrones principales sin sobreajustar los datos.

```
In [1]: # Calcular el R2  
observed <- Lst23(fd[, 1])  
predicted <- model.funcional$yhatfdobj
```

rmse <- sqrt(mean((observed - predicted)^2))

print(paste("RMSE del modelo:", round(rmse, 3)))

[1] "RMSE del modelo: 2.541"

El RMSE igual a 2.541, nos sugiere un modelo razonablemente preciso

```
In [1]: names(model.funcional$betaestlist)
```

"const" "STID.NVDI"

'done'

Coeficiente asociado a STID(NDVI)



Podemos observar que cuando beta toma su valor más negativo, es en el mes de junio, coincidiendo con el inicio del verano, cuando las temperaturas suelen ser las más altas.

Respuesta funcional $LST_{23}(t)$ y predictor funcional NDVI₂₃(t), donde $LST_{23}(t) = \text{NDVI}(t+60)$ con $t = 1, \dots, 12$.

```
In [1]: # Seleccionamos los últimos 12 meses  
Lst23 <- as.matrix(Dlst2[61:72, ])
```

Buscamos el numero de base, con el cual no haya un sobreajuste

nbasis.v <- c(10, 15, 20, 25, 30, 35, 40, 50, 60)

Suponemos y graficar para cada valor de nbasis

par(mfrow = c(2, 2)) # Configuración para múltiples gráficos

for (nbasis in nbasis.v) {
 Lst23.basis <- create.fourier.basis(rangeval = c(1, 12), nbasis = nbasis)

NDVI23.fd <- smooth.basis(1:t, Lst23, Lst23.basis)\$fd

Graficar NDVI₂₃(t) para todos los sitios

plot(NDVI23.fd, main = paste("nbasis =", nbasis))

'done'

Curvas observadas vs predichas

De acuerdo a estas graficas, seleccionamos nbasis igual a 6, pues captura adecuadamente las tendencias estacionales a lo largo de los meses, de igual forma no refleja ruido.

```
In [1]: Lst23.basis <- create.fourier.basis(rangeval = c(1, 12), nbasis = 6)
```

NDVI23.fd <- create.fourier.basis(rangeval = c(1, 12), nbasis = 6)

NDVI23.fd <- smooth.basis(1:t, NDVI23, NDVI23.basis)\$fd

```
In [1]: # Ajustamos el modelo funcional
```

model.funcional <- fRegress(Lst23.fd - NDVI23.fd)

'done'

Coeficiente funcional asociado a NDVI23

Podemos observar que cuando beta toma su valor más negativo, es en el mes de junio, coincidiendo con el inicio del verano, cuando las temperaturas suelen ser las más altas.

Curvas observadas y predichas están bien alineadas, lo que indica un buen ajuste general del modelo.

```
In [1]: # Predicciones del modelo funcional
```

Lst23.pred <- model.funcional\$yhatfdobj

Gráfica de comparación entre valores observados y predichos