

Tema 8. Proyecto 3

Viacheslav Shalisko

9 de abril de 2019

Interpolación de residuales para modelo de predicción de temperaturas en el Occidente de México

Cargar las bibliotecas requeridas

```
library(sp)
library(raster)
library(rgdal)
```

```
## rgdal: version: 1.4-4, (SVN revision 833)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 2.2.3, released 2017/11/20
## Path to GDAL shared files: C:/Users/vshal/Documents/R/win-library/3.6/rgdal/gdal
## GDAL binary built with GEOS: TRUE
## Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
## Path to PROJ.4 shared files: C:/Users/vshal/Documents/R/win-library/3.6/rgdal/proj
## Linking to sp version: 1.3-1
```

```
library(foreign)
library(RColorBrewer)
#library(rasterVis)
library(mgcv)
```

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

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:raster':
##
##   getData
```

```
## This is mgcv 1.8-28. For overview type 'help("mgcv-package")'.
```

Definir variables

```
# definir área de trabajo
mi_recorte <- extent(c(-110, -100, 17, 23))

escala_t <- seq(8, 32, by=1)

# definir colores para temperatura
mis_colores <- colorRampPalette(c("blue","lightyellow","red"))(length(escala_t)-1)
mis_colores_residuales <- colorRampPalette(c("coral", "lightblue"))(10)

# nombre de capa raster con mdelo paramétrico de temperatura (sin extensión)
nombre_raster_t <- "prediccion_tjunio_32bits"
# nombre de la columna en datos SMN con valores de temperatura correctos
nombre_columna_t <- "TJUN"
```

Lectura de los archivos fuente

1. GeoTIFF
2. Shapefile de puntos (SHP)
3. Tabla DBF

```
# raster con datos del modelo paramétrico
temp_parametrico <- raster("datos/prediccion_tjunio_32bits.tif")
temp_parametrico
```

```
## class      : RasterLayer
## dimensions : 1920, 4865, 9340800  (nrow, ncol, ncell)
## resolution : 0.008333333, 0.008333333  (x, y)
## extent     : -108.5249, -67.98321, 9.915776, 25.91578  (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
## source     : C:/Users/vshal/GD/UdeG_Docencia/CUCSH_Curso_R/sources/datos/prediccion_tjunio_32bits.tif
## names      : prediccion_tjunio_32bits
## values     : 4.033294, 29.28952  (min, max)
```

```
# puntos SMN
smn <- readOGR("datos/Estaciones_SMN_alt_ok.shp")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\vshal\GD\UdeG_Docencia\CUCSH_Curso_R\sources\datos\Estaciones_SMN_alt_ok.shp", layer: "Estaciones_SMN_alt_ok"
## with 3089 features
## It has 10 fields
```

```
smn
```

```
## class      : SpatialPointsDataFrame
## features   : 3089
## extent     : -117.0472, -86.82028, 14.61778, 32.665 (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
## variables  : 10
## names      : OBJECTID, NUMERO_DE_,          NOMBRE_DE_,      Z, CVE_ENT,  Z_DELTA,  Z_M
DE, Z_MDE1, Z_DELTA1,  Z_FIN
## min values :      1,    00001003,          ?ADO - ACULCO,    0,      01, -382.393,    -
14,      -14, -382.393,      -14
## max values :    3089,    00032187, ZUMPANGO DEL RIO (SMN), 4110,      32,  313.404, 4297.
51, 4297.51,  313.404, 4297.51
```

```
# tabla de temperaturas SMN
temp <- read.dbf("datos/Temp.dbf")
head(temp)
```

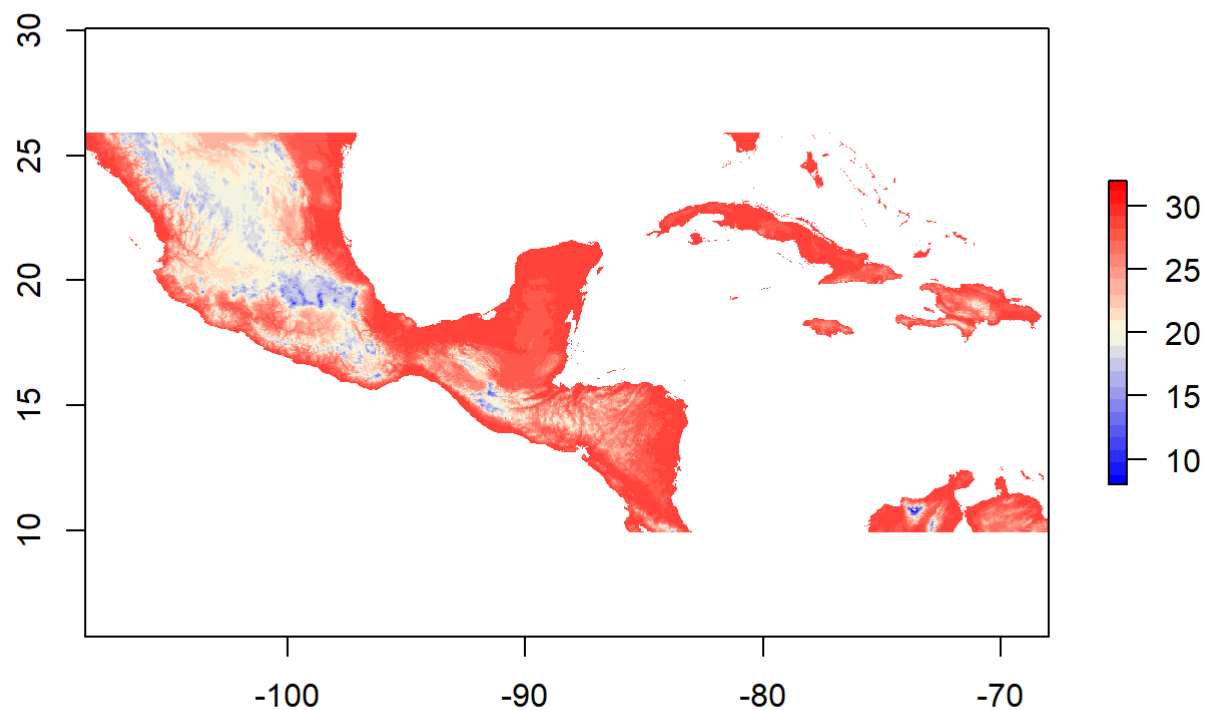
```
##  NUMERO_DE_  TENE TFEB TMAR TABR TMAY TJUN TJUL TAGO TSEP TOCT TNOV TDIC
## 1    00001003 15.4 16.7 19.2 21.7 23.7 24.0 22.0 22.1 22.2 21.0 18.5 16.3
## 2    00001004 11.8 13.1 15.6 18.0 19.9 20.7 19.4 19.2 18.6 16.7 14.2 12.5
## 3    00001005 13.4 14.7 17.2 19.5 21.9 22.6 21.0 20.7 20.2 18.6 16.2 14.2
## 4    00001006 12.6 13.8 16.7 19.5 21.7 22.0 20.5 20.3 19.8 18.0 15.1 13.4
## 5    00001007 12.2 13.4 16.4 18.8 21.0 21.4 20.0 19.6 19.5 17.5 14.8 13.1
## 6    00001008 13.0 13.8 16.0 18.1 20.0 19.5 18.3 18.3 17.6 16.4 14.9 13.4
##  TMEDIA
## 1    20.2
## 2    16.6
## 3    18.4
## 4    17.8
## 5    17.3
## 6    16.6
```

Visualización y recorte de datos fuente (raster de temperaturas estimados con el modelado paramétrico)

```
temp_parametrico_recorte <- crop(temp_parametrico, mi_recorte)

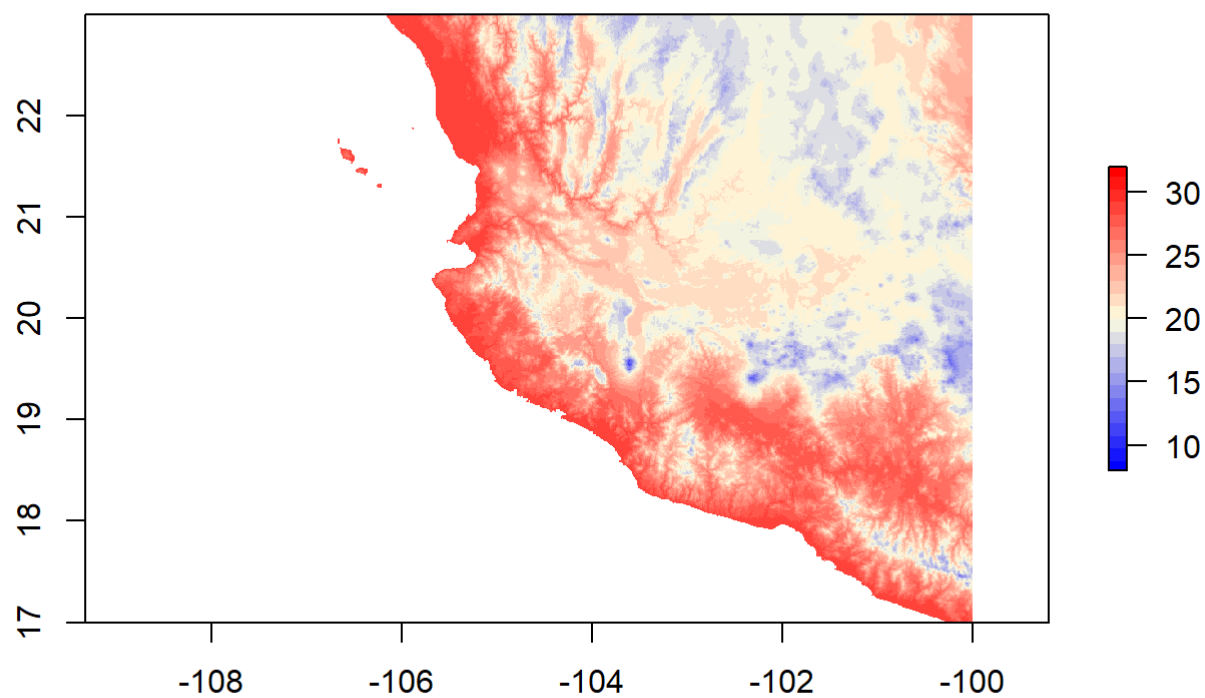
plot(temp_parametrico, zlim = c(8, 32), col = mis_colores, main = "antes de recorte")
```

antes de recorte



```
plot(temp_parametrico_recorte, zlim = c(8, 32), col = mis_colores, main = "despues de recorte")
```

despues de recorte



Vincular puntos SMN y datos de temperatura (JOIN)

```
smn <- merge(smn, temp, by = "NUMERO_DE_")
smn
```

```
## class      : SpatialPointsDataFrame
## features    : 3089
## extent      : -117.0472, -86.82028, 14.61778, 32.665 (xmin, xmax, ymin, ymax)
## crs         : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
## variables   : 23
## names       : NUMERO_DE_, OBJECTID,          NOMBRE_DE_,    Z, CVE_ENT,  Z_DELTA,  Z_M
DE, Z_MDE1, Z_DELTA1,  Z_FIN, TENE, TFEB, TMAR, TABR, TMAY, ...
## min values  : 00001003,      1,          ?ADO - ACULCO,    0,      01, -382.393,    -
14,      -14, -382.393,      -14,    0,    0,    0,    0,    0, ...
## max values  : 00032187,      3089, ZUMPANGO DEL RIO (SMN), 4110,      32,  313.404, 4297.
51, 4297.51,  313.404, 4297.51, 28.7, 28.9, 30.8, 32.7, 33.6, ...
```

Realizar muestreo de predicción del modelo paramétrico en los puntos de SMN

Vincular resultados de muestreo con capa de pntos y depurar

```
# muestreo eon todos puntos SMN (como data.frame)
temp_parametrico_muestreo <- extract(temp_parametrico_recorte, smn, df = TRUE)

# vinculación de los resultados de muestreo con el conjunto de puntos (su tabla de atributos)
smn@data <- cbind(smn@data, temp_parametrico_muestreo)

# revisar capa de datos SMN despues de viclular columnas de muestreo
smn
```

```
## class      : SpatialPointsDataFrame
## features    : 3089
## extent      : -117.0472, -86.82028, 14.61778, 32.665 (xmin, xmax, ymin, ymax)
## crs         : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
## variables   : 25
## names       : NUMERO_DE_, OBJECTID,          NOMBRE_DE_,    Z, CVE_ENT,  Z_DELTA,  Z_M
DE, Z_MDE1, Z_DELTA1,  Z_FIN, TENE, TFEB, TMAR, TABR, TMAY, ...
## min values  : 00001003,      1,          ?ADO - ACULCO,    0,      01, -382.393,    -
14,      -14, -382.393,      -14,    0,    0,    0,    0,    0, ...
## max values  : 00032187,      3089, ZUMPANGO DEL RIO (SMN), 4110,      32,  313.404, 4297.
51, 4297.51,  313.404, 4297.51, 28.7, 28.9, 30.8, 32.7, 33.6, ...
```

```
# seleccionar solo puntos SMN con resultados de muestreo
mascara_puntos_muestreados <- !is.na(smn@data[,nombre_raster_t])
smn_selectos <- smn[mascara_puntos_muestreados,]

# descartar registros con temperaturas incorrectas (T = 0)
mascara_temp_ok <- smn_selectos@data[,nombre_columna_t] != 0
smn_selectos <- smn_selectos[mascara_temp_ok,]

# revisar capa de puntos SMN selectos
smn_selectos
```

```
## class      : SpatialPointsDataFrame
## features   : 700
## extent     : -105.85, -100, 17.00222, 22.98333 (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
## variables  : 25
## names      : NUMERO_DE_, OBJECTID,          NOMBRE_DE_,    Z, CVE_ENT,  Z_DELTA,
Z_MDE,      Z_MDE1, Z_DELTA1,    Z_FIN, TENE, TFEB, TMAR, TABR, TMAY, ...
## min values : 00001003,      1,          ABASOLO - ABASOLO,    0,      01, -382.349, -
0.831517, -0.831517, -382.349, -0.831517, 7.3, 8.5, 10.5, 12.1, 13.4, ...
## max values : 00032187,      3089, ZIRITZICUARO - LA HUACANA, 2930,      32, 313.404,
2962.18, 2962.18, 313.404, 2962.18, 27.8, 28.8, 30.6, 32.7, 33.6, ...
```

```
head(smn_selectos@data)
```

```
##  NUMERO_DE_ OBJECTID          NOMBRE_DE_    Z CVE_ENT    Z_DELTA
## 1  00001003      1      CALVILLO - CALVILLO 1702      01 -0.666382
## 2  00001004      2 CA?ADA HONDA - AGUASCALIE 1925      01 8.635380
## 3  00001005      3 PRESA EL NIAGARA - AGUASC 1828      01 -75.760100
## 4  00001006      4      EL TULE - ASIENTOS 1970      01 -30.881600
## 5  00001007      5 JESUS MARIA - JESUS MARIA 1800      01 -85.542100
## 6  00001008      6 PUERTO DE LA CONCEPCION - 2300      01 -22.656500
##    Z_MDE  Z_MDE1  Z_DELTA1  Z_FIN TENE TFEB TMAR TABR TMAY TJUN TJUL
## 1 1702.67 1702.67 -0.666382 1702.67 15.4 16.7 19.2 21.7 23.7 24.0 22.0
## 2 1916.36 1916.36 8.635380 1916.36 11.8 13.1 15.6 18.0 19.9 20.7 19.4
## 3 1903.76 1903.76 -75.760100 1903.76 13.4 14.7 17.2 19.5 21.9 22.6 21.0
## 4 2000.88 2000.88 -30.881600 2000.88 12.6 13.8 16.7 19.5 21.7 22.0 20.5
## 5 1885.54 1885.54 -85.542100 1885.54 12.2 13.4 16.4 18.8 21.0 21.4 20.0
## 6 2322.66 2322.66 -22.656500 2322.66 13.0 13.8 16.0 18.1 20.0 19.5 18.3
##    TAGO TSEP TOCT TNOV TDIC TMEDIA ID prediccion_tjunio_32bits
## 1 22.1 22.2 21.0 18.5 16.3 20.2 1 21.11999
## 2 19.2 18.6 16.7 14.2 12.5 16.6 2 20.15940
## 3 20.7 20.2 18.6 16.2 14.2 18.4 3 20.14580
## 4 20.3 19.8 18.0 15.1 13.4 17.8 4 19.73801
## 5 19.6 19.5 17.5 14.8 13.1 17.3 5 20.26814
## 6 18.3 17.6 16.4 14.9 13.4 16.6 6 18.43306
```

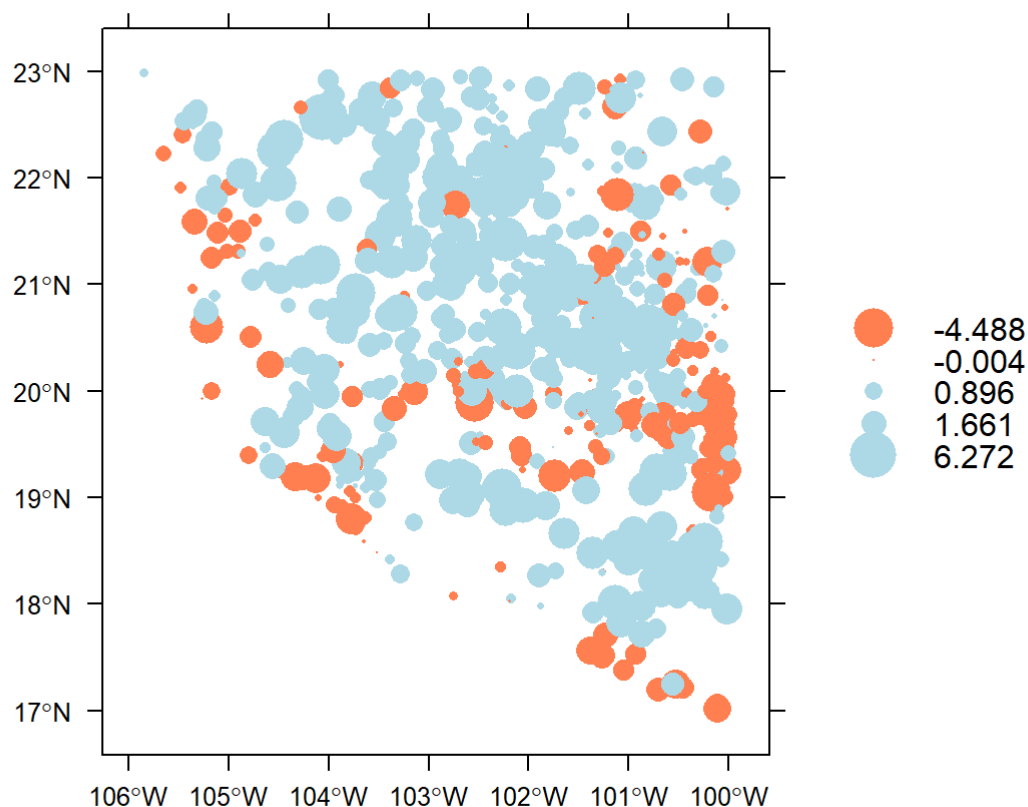
```
# visualizar puntos SMN antes y despues de selecci3n
plot(smn, col = "lightgray")
plot(smn_selectos, col = "black", add = TRUE)
```



Calcular y visualizar los residuales

```
# calcular residuales: de valores observados restar los valores estimados con el modelo param  
étrico  
smn_selectos@data$residuales <- smn_selectos@data[,nombre_columna_t] - smn_selectos@data[,nom  
bre_raster_t]  
  
# visualizar residuales  
bubble(smn_selectos, zcol='residuales', scales = list(draw = TRUE),  
        col = c("coral", "lightblue"), main = "Distribución de residuales",  
        key.space = list("right"))
```

Distribución de residuales



Construir un modelo GAM para interpolación de residuales

```
# colocar datos en variables con nombres cortos
lon <- smn_selectos@coords[,1]
lat <- smn_selectos@coords[,2]
res <- smn_selectos@data$residuales

interp_resid <- gam(res ~ s(lon, lat, k = 150))

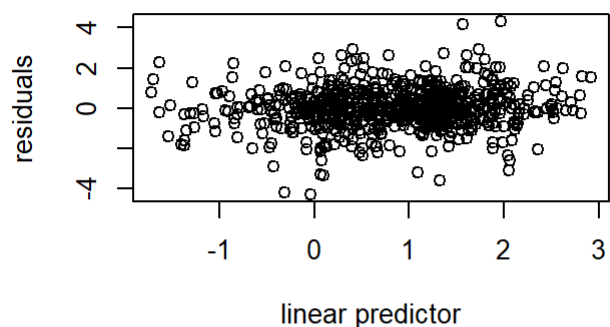
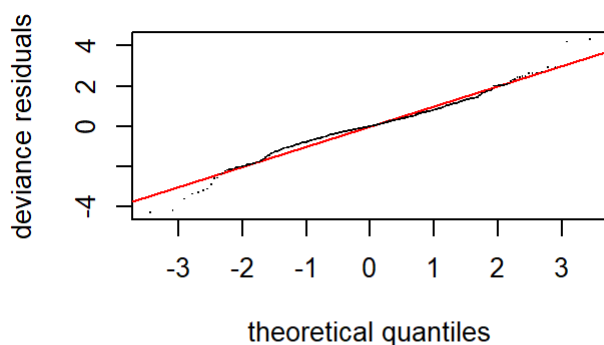
summary(interp_resid)
```



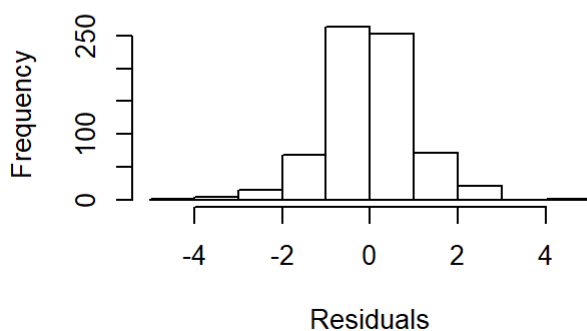
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## res ~ s(lon, lat, k = 150)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.83416    0.04078   20.46  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df    F p-value
## s(lon,lat) 76.86   102 4.871  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.413   Deviance explained = 47.8%
## GCV = 1.3096   Scale est. = 1.164       n = 700
```

```
gam.check(interp_resid)
```

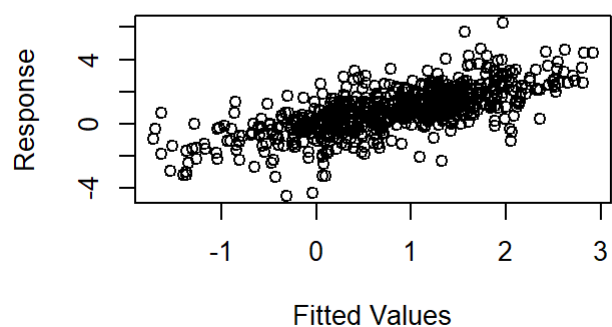
Resids vs. linear pred.



Histogram of residuals

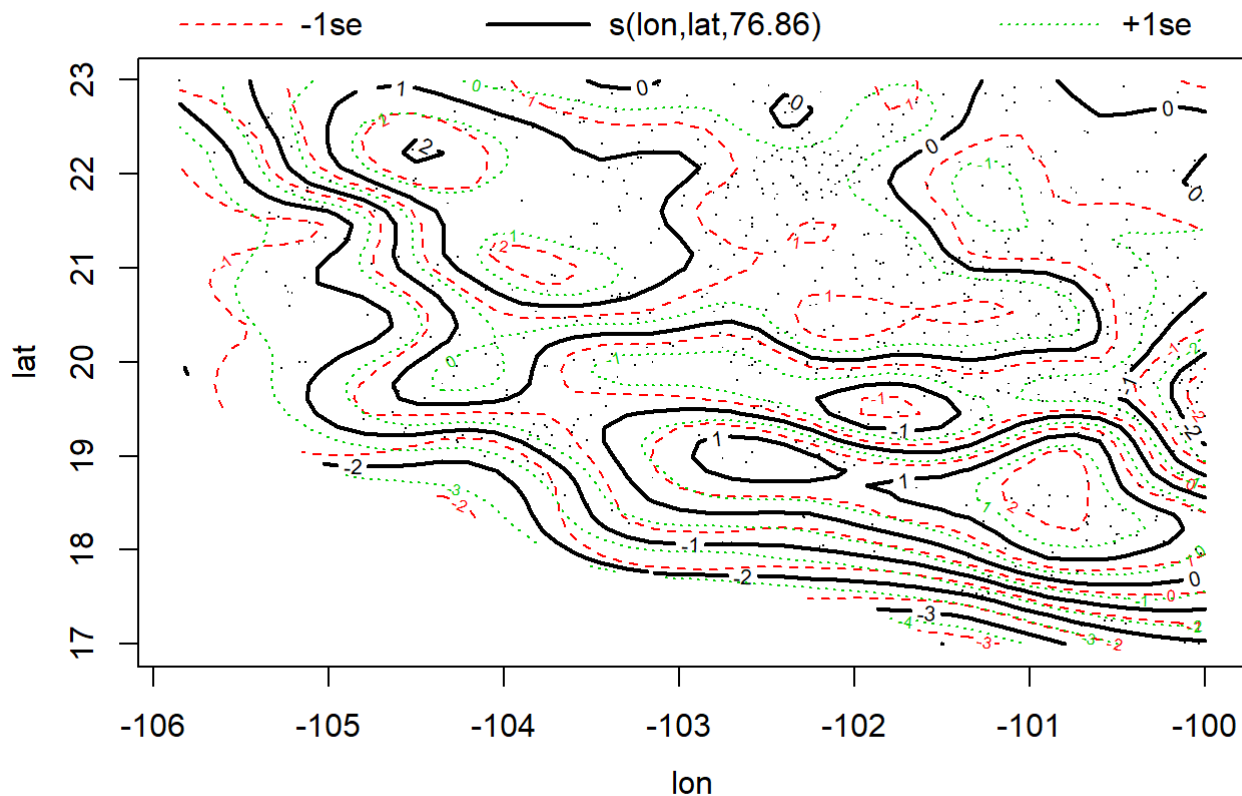


Response vs. Fitted Values



```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 4 iterations.
## The RMS GCV score gradient at convergence was 2.570934e-09 .
## The Hessian was positive definite.
## Model rank = 150 / 150
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##           k'   edf k-index p-value
## s(lon,lat) 149.0 76.9   1.08     1
```

```
plot(interp_resid, n = 1000)
```



Generar raster de superficie de ajuste de temperaturas

El proceso de interpolación puede durar algunos minutos (depende del tamaño de área de interpolación)

```
# generar capas con valores numericos de latitud y longitud
r_X <- init(temp_parametrico_recorte, 'x')
r_Y <- init(temp_parametrico_recorte, 'y')

# colocar capas de latitud y longitud en un solo objeto raster
# los nombres de capas deben ser mismos que variables en GAM
r_latlon <- stack(r_X,r_Y)
names(r_latlon) <- c("lon","lat")

r_latlon
```

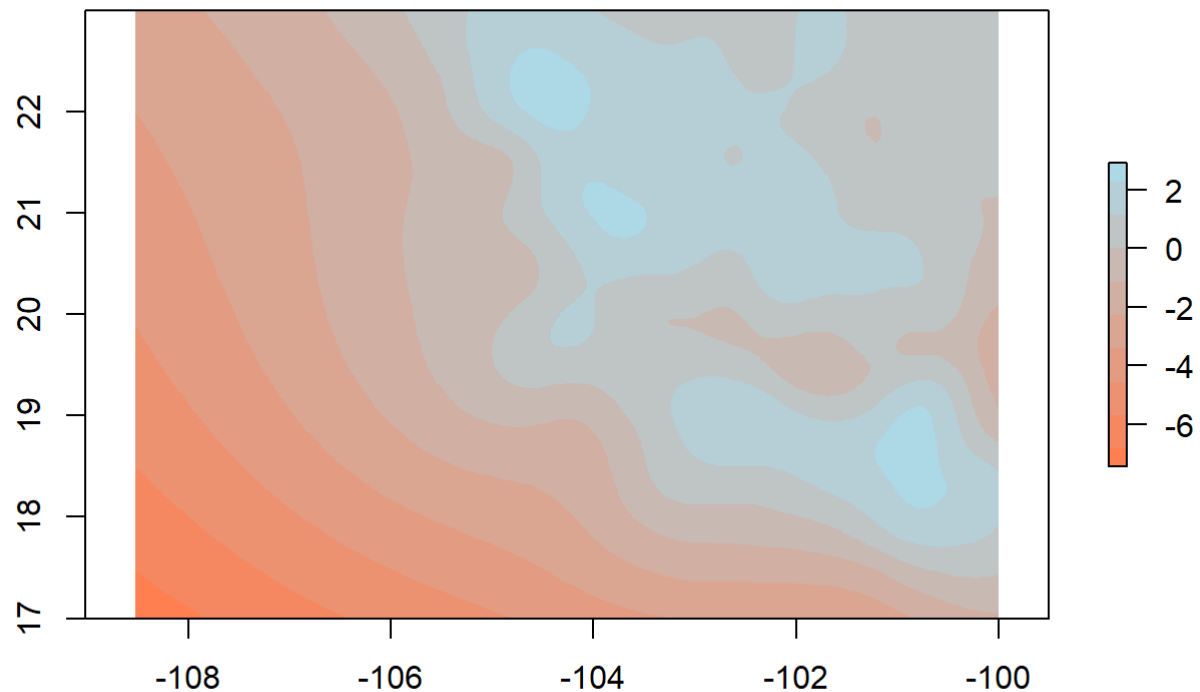
```
## class      : RasterStack
## dimensions : 720, 1023, 736560, 2  (nrow, ncol, ncell, nlayers)
## resolution : 0.008333333, 0.008333333  (x, y)
## extent     : -108.5249, -99.99988, 16.99911, 22.99911  (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
## names      :      lon,      lat
## min values : -108.52071,  17.00328
## max values : -100.00404,  22.99494
```

```
# generar superficie de residuales (paso lento)
raster_residuales <- predict(r_latlon, interp_resid)

raster_residuales
```

```
## class      : RasterLayer
## dimensions : 720, 1023, 736560  (nrow, ncol, ncell)
## resolution : 0.008333333, 0.008333333  (x, y)
## extent     : -108.5249, -99.99988, 16.99911, 22.99911  (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
## source     : memory
## names      : layer
## values     : -7.452167, 2.924659  (min, max)
```

```
plot(raster_residuales, asp = 1, col = mis_colores_residuales)
```

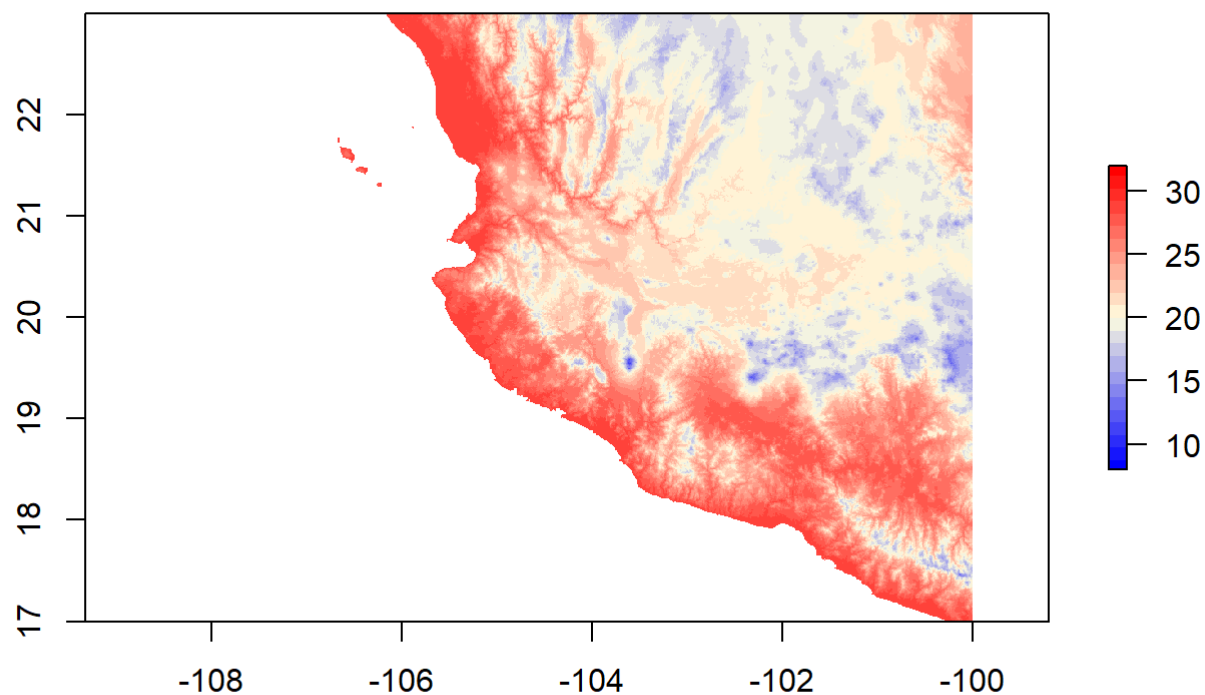


Realizar corrección de raster de temperaturas

```
# sumar la superficie interpolada con el modelo paramétrico y residuales interpolados
temp_corregido <- temp_parametrico_recorte + raster_residuales

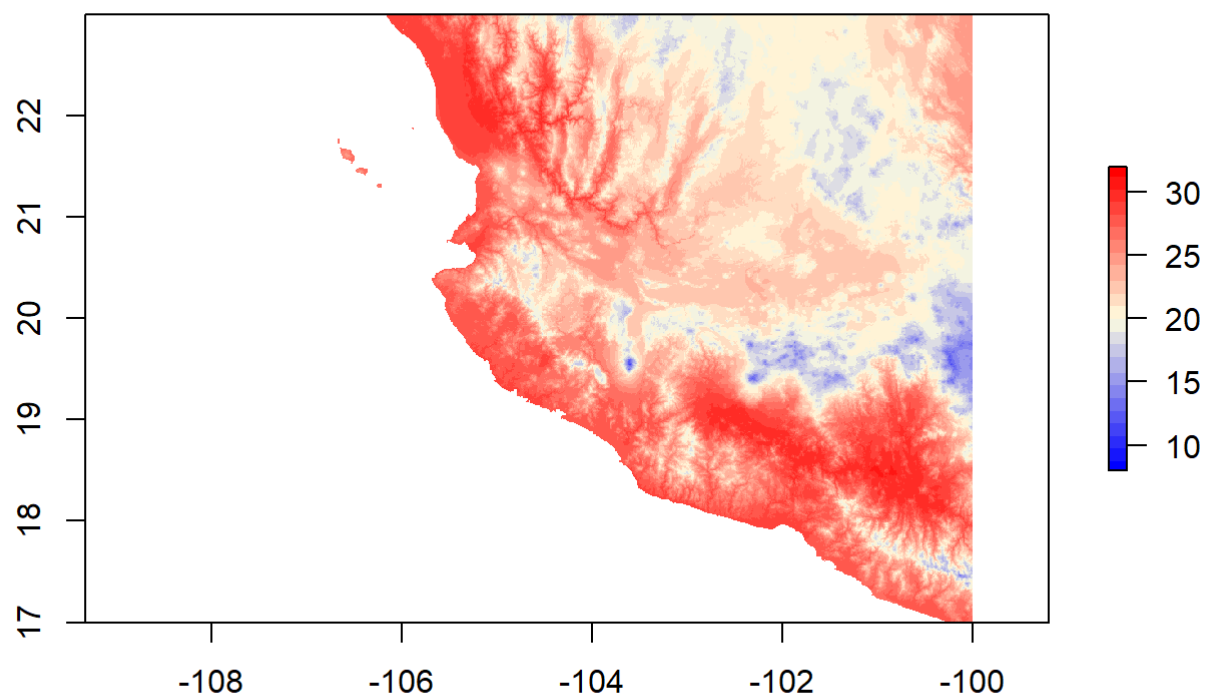
plot(temp_parametrico_recorte, zlim = c(8, 32), col = mis_colores, main = "Temperaturas sin c
orrección")
```

Temperaturas sin corrección



```
plot(temp_corregido, zlim = c(8, 32), col = mis_colores, main = "Temperaturas despues de corrección")
```

Temperaturas despues de corrección



Guardar raster de resultados de corrección en GeoTIFF (Float 32 bits)

```
writeRaster(temp_corregido, filename = "datos/corregido_tjunio_32bits.tif",  
            format="GTiff", datatype = "FLT4S", overwrite = TRUE)
```