Maestría en Estadística Aplicada y Ciencia de Datos.

Expresiones regulares e imagenes satelitales

Stalyn Guerrero
Docente UNBOSQUE

2022-06-17

1. Expresiones regulares

Introducción.

Para realizar este proceso debemos conectarnos a diferentes repositorios de imagenes desde *R*.

Primero pasos en earthengine.google.com

1. Crear una cuenta en earthengine.google.com, una vez que se ingrese a la cuenta puede buscarse los conjuntos de datos de interés, por ejemplo:
   - DMSP OLS: Nighttime Lights Time Series Version 4, Defense Meteorological Program Operational Linescan System
   - Copernicus Global Land Cover Layers: CGLS-LC100 Collection 3
   - CSP gHM: Global Human Modification

Instalación de rgee

- Descargar e instalar anaconda o conda. (https://www.anaconda.com/products/individual)
- Abrir Anaconda prompt y configurar ambiente de trabajo (ambiente python rgee_py) con las siguientes sentencias:

```
conda create -n rgee_py python=3.9

activate rgee_py

pip install google-api-python-client

pip install earthengine-api

pip install numpy
```

Validar los ambientes.

- Listar los ambientes de Python disponibles en anaconda `prompt`

  `conda env list`

- Guía MAC
- Una vez identificado la ruta del ambiente ambiente rgee_py definirla en `R` , Instalar reticulate y rgee, cargar paquetes para procesamiento espacial y configurar el ambiente de trabajo como sigue:

Probemos R

```
library(reticulate) # Conexión con Python
library(rgee) # Conexión con Google Earth Engine
library(sf) # Paquete para manejar datos geográficos
library(concaveman)
library(geojsonio)
library(magrittr)
```

Configuración inicial de Python

```
rgee_environment_dir <-
  "C:/Users/guerr/.conda/envs/rgee_py/python.exe"
# Configurar python (Algunas veces no es detectado y se debe reiniciar R)
reticulate::use_python(rgee_environment_dir, required = T)

rgee::ee_install_set_pyenv(py_path = rgee_environment_dir, py_env = "rgee_py")

Sys.setenv(RETICULATE_PYTHON = rgee_environment_dir)

Sys.setenv(EARTHENGINE_PYTHON = rgee_environment_dir)

rgee::ee_Initialize(drive = T)
```

Descargando imagenes.

- Una vez realizado el proceso de configurar `R`, `Python` y **Google Earth Engine,** estamos listos para extraer información auxiliar a partir de la información Satelital.
- Las imágenes satélitales se procesan a partir de los polígonos del país, por lo cual es primer paso es hacer la lectura de la shapefiles.

```
COL <- read_sf("../mpio/mpio.shp")
```

Procesando la información auxiliar de **luces nocturnas**.

```
luces <- ee$ImageCollection("NOAA/DMSP-OLS/NIGHTTIME_LIGHTS") %>%
  ee$ImageCollection$filterDate("2013-01-01", "2014-01-01") %>%
  ee$ImageCollection$map(function(x) x$select("stable_lights")) %>%
  ee$ImageCollection$toBands()
ee_print(luces)

COL_luces <- ee_extract(
  x = luces,
  y = COL["MPIO"],
  ee$Reducer$mean(),
  sf = FALSE
)
```

Nota: La sintaxis mostrada previamente es una pequeña variación de la disponible en la pagina web de las imágenes.

Section 1

Expresiones regulares

Expresiones regulares

- Es un lenguaje formal para denotar o describir un lenguaje regula.
- El lenguaje es un conjunto de cadenas, entonces las expresiones regulares denotan un conjunto de cadenas pertenecientes a un lenguaje regular.

Definición

Una expresión regular (RE, siglas en inglés) es un mecanismo que permite seleccionar una cadena o sarta específica de otra cadena de caracteres.

Metacarácters especiales

Algunos elementos que no son metacarácteres, pero que se utilizan como una secuencia de “escape”.

- **a** Representa una “campana” o “beep” que se produce al imprimir este carácter.
- **e** Representa la tecla “Esc” o “Escape”,
- **n** Representa un salto de linea,
- **r** Representa el “retorno de carro” o “regreso al inicio” o sea el lugar en que la línea vuelve a iniciar
- **t** Representa un tabulador.
- **v** Representa un tabulador vertical
- **f** Representa un salto de página
- **!** Representa una busqueda negativa, en otras palabras que no incluya la palabra que especificamos.

Barra invertida.

Carácter	Significado
\w	Indica alfanumérico
\W	No Alfanumérico
\d	Indica Numérico
\D	No numérico
\s	Indica Espacio
\S	No espacio
\	Genera un escape para darle significado literal a un metacarácter.
\A	Representa el inicio de la cadena. No un carácter sino una posición.
\Z	Representa el final de la cadena. No un carácter sino una posición.
\b	Marca la posición de una palabra limitada por espacios en blanco, puntuación o el inicio/final de una cadena.
\B	Marca la posición entre dos caracteres alfanuméricos o

Ejemplo de barras

```
x <- c(
"¿Con que frecuencia consumió Queso, kumis, yogurt, queso crem
"Usualmente, en UN mes, ¿(...) consume:Granos secos (fríjol, a
"16. ¿Con que frecuencia consumió Granos secos (fríjol, arveja
"17. Usualmente, en UN mes, ¿(...) consume:Arroz o pasta?",
"¿Con que frecuencia consumió Arroz o pasta?",
"18. Usualmente, en UN mes, ¿(...) consume:Pan?"
)

gsub(x =  x, pattern =  "\\D", replacement = "" )
```

```
## [1] ""   ""   "16" "17" ""   "18"
```

```
gsub(x =  x, pattern =  "\\d", replacement = "" )
```

```
## [1] "¿Con que frecuencia consumió Queso, kumis, yogurt, que
## [2] "Usualmente, en UN mes, ¿(...) consume:Granos secos (fr
```

Ejemplo de barras

```
gsub(x =  x, pattern =  "\\S", replacement = "" )
```

```
## [1] "                " "                " "                " "
## [6] "      "
```

```
gsub(x =  x, pattern =  "\\s", replacement = "" )
```

```
## [1] "¿ConquefrecuenciaconsumióQueso,kumis,yogurt,quesocrema
## [2] "Usualmente,enUNmes,¿(...)consume:Granossecos(fríjol,ar
## [3] "16.¿ConquefrecuenciaconsumióGranossecos(fríjol,arveja,
## [4] "17.Usualmente,enUNmes,¿(...)consume:Arrozopasta?"
## [5] "¿ConquefrecuenciaconsumióArrozopasta?"
## [6] "18.Usualmente,enUNmes,¿(...)consume:Pan?"
```

```
gsub(x =  x, pattern =  "kumis", replacement = "_____" )
```

```
## [1] "¿Con que frecuencia consumió Queso, _____, yogurt, que
## [2] "Usualmente, en UN mes, ¿(...) consume:Granos secos (fr
```

Cuantificadores.

Un Cuatinficardor que precede a un carácter cuantifica las veces que este carácter puede aparecer.

- **?** El carácter que precede puede aparecer como mucho una vez.
- **+** El carácter que le precede debe aparecer al menos una vez.
- ***** El carácter que le precede puede aparecer cero, una, o más veces.
- **{}** Las llaves juegan el papel de meta caracteres, para que cumplan su funcionalidad deben estar después de la expresión regular y encierren uno o varios números.
  - **{n}** Indica que coincide *n* veces.
  - **{n,}** Indica que coincide mas *n* veces.
  - **{,n}** Indica que coincide hasta *n* veces.
  - **{n,m}** Indica que coincide mas *n* veces y menos de *m* veces.
- **$** El signo de pesos representa el final de la cadena de caracteres o el final de la línea.
- **^** El gorro representa el inicio de la cadena.

Cuantificadores

```
gsub(x =  x, pattern =  "?\\.", replacement = "(-_-)" )
```

```
## [1] "¿Con que frecuencia consumió Queso, kumis, yogurt, que
## [2] "Usualmente, en UN mes, ¿((-_-)(-_-)(-_-)) consume:Gran
## [3] "16(-_-) ¿Con que frecuencia consumió Granos secos (fri
## [4] "17(-_-) Usualmente, en UN mes, ¿((-_-)(-_-)(-_-)) cons
## [5] "¿Con que frecuencia consumió Arroz o pasta?"
## [6] "18(-_-) Usualmente, en UN mes, ¿((-_-)(-_-)(-_-)) cons
```

```
gsub(x =  x, pattern =  "?\\.+", replacement = "-_-" )
```

```
## [1] "¿Con que frecuencia consumió Queso, kumis, yogurt, que
## [2] "Usualmente, en UN mes, ¿(-_-) consume:Granos secos (fr
## [3] "16-_- ¿Con que frecuencia consumió Granos secos (fríjo
## [4] "17-_- Usualmente, en UN mes, ¿(-_-) consume:Arroz o pa
## [5] "¿Con que frecuencia consumió Arroz o pasta?"
## [6] "18-_- Usualmente, en UN mes, ¿(-_-) consume:Pan?"
```

Agrupación

- **()** Los parentesis son usados para la aplicación de operadores sobre mas de un carácter.
- **[]** Los corchetes agrupan caracteres en grupos o clases. Son útiles cuando es necesario buscar uno de un grupo de caracteres.
- **[a-z]** Especifica un rango de caracteres.
- **[^....]** Lista de caracteres excluidos
- | Una barra vertical separa las alternativas. Realiza el papel de o.
- **.** El punto busca cualquier carácter sin incluir los saltos de línea.

Agrupación

```
gsub(x =  x, pattern =  "[A-Z]", replacement = "(-_-)" )
```

```
## [1] "¿(-_-)on que frecuencia consumió (-_-)ueso, kumis, yog
## [2] "(-_-)sualmente, en (-_-)(-_-) mes, ¿(...) consume:(-_-
## [3] "16. ¿(-_-)on que frecuencia consumió (-_-)ranos secos
## [4] "17. (-_-)sualmente, en (-_-)(-_-) mes, ¿(...) consume:
## [5] "¿(-_-)on que frecuencia consumió (-_-)rroz o pasta?"
## [6] "18. (-_-)sualmente, en (-_-)(-_-) mes, ¿(...) consume:
```

```
gsub(x =  x[1], pattern =  "[a-z]", replacement = "_" )
```

```
## [1] "¿C__ ___ _________ _______ó Q____, _____, ______, ___
```

```
gsub(x =  x[1], pattern =  "[á|é|í|ó|ú|ñ]", replacement = "__
```

```
## [1] "¿Con que frecuencia consumi____ Queso, kumis, yogurt,
```

```
gsub(x =  x[3], pattern =  "(\\(.*\\))", replacement = "_____"
```

Ejemplos varios

Las funciones más usuales que trabajan con expresiones regulares en R son:

grep, **grepl**; son funciones usadas para búsquedas de coincidencias:

```
String <- c("regular", "expression", "example", "of", "R", "le
grep(pattern = "ex", x = String, value = TRUE)
```

```
## [1] "expression" "example"
```

```
grep(pattern = "ex", x = String, value = FALSE)
```

```
## [1] 2 3
```

```
grepl(pattern = "ex", x = String)
```

```
## [1] FALSE  TRUE  TRUE FALSE FALSE FALSE
```

Ejercicios

Empleando la base SB11_20141 de la librería saber responda las siguientes preguntas:

Solución

1. ¿Cuantos código DANE tiene 4 ceros seguidos?

```
require(saber)
data("SB11_20141")
COD_DENE <- unique(SB11_20141$CODIGO_DANE)
grep(pattern = "0{4}", x = COD_DENE, value = TRUE)
grep(pattern = "0000", x = COD_DENE, value = TRUE)
```

2. ¿Cuantos códigos DANE terminar en número par antecedidos por el número 7?

```
grep(pattern = "7[24680]$", x = COD_DENE, value = TRUE)
```

3. ¿Cuantos códigos DANE inician por impar seguidos del numero 2?

```
grep(pattern = "^[13579]2", x = COD_DENE, value = TRUE)
```

4. ¿Cuantos códigos DANE inician por 2 o 3 y el cuarto dígito sea 7 o 9 y termine en 13