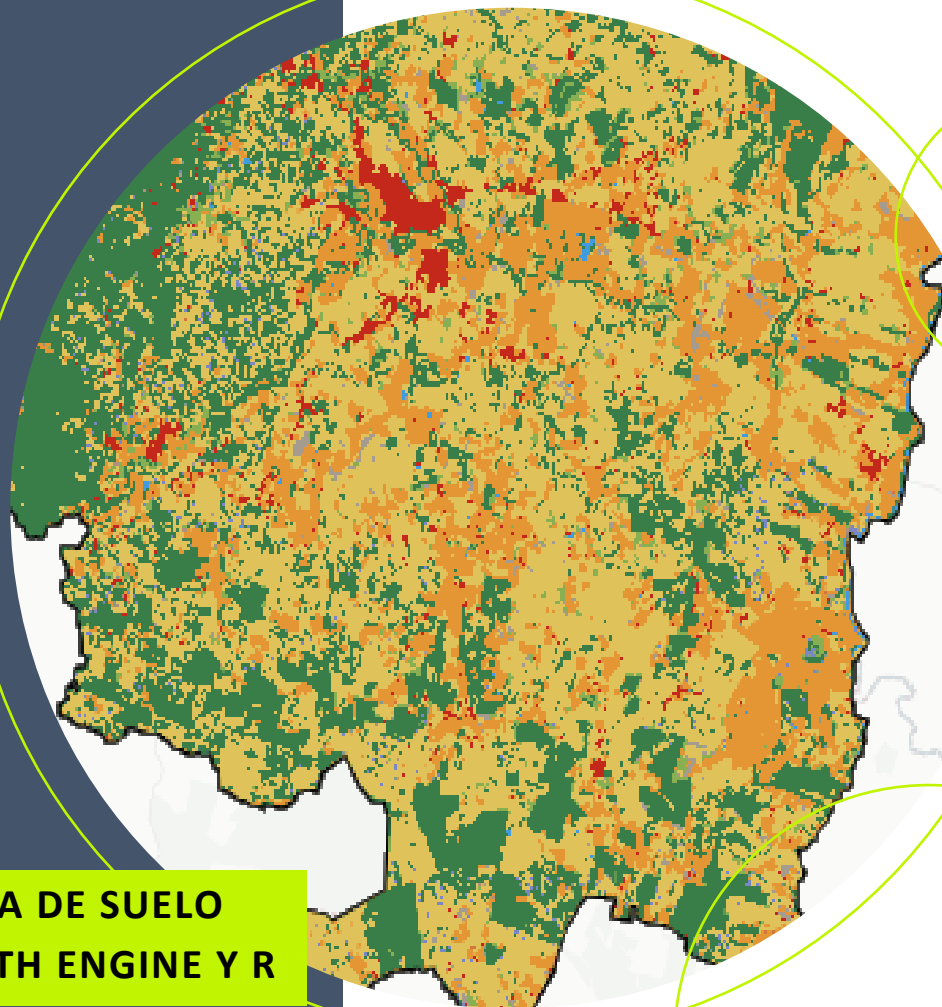




GUIA



COBERTURA DE SUELO USANDO EARTH ENGINE Y R

Felipe Muñoz S.

Desarrollador.

felipe.munoz1601@alumnos.ubiobio.cl

Bastían Candia C.

Desarrollador.

bastian.candia1801@alumnos.ubiobio.cl

Clemente Rubio M.

Profesor Guía.

clrubio@ubiobio.cl

CONTEXTO

Como parte del curso anteproyecto y como proyecto del electivo de R creamos este script para extraer y graficar el tipo de cobertura de suelo en cualquier parte del mundo a partir de junio del 2015.

La zona geográfica que se usa para esta guía es la provincia de Cauquenes ubicada en la región del Maule.

Los Raster provenientes del script son compatibles con programas de geo informática como lo son **QGIS** y **Dinámica EGO**. Así, se pueden realizar análisis y proyecciones de escenarios futuros usando dichos rasters.

DYNAMIC WORLD V1

Dynamic World V1 es un dataset disponible en el catálogo de Earth Engine, en él se encuentran las probabilidades de cobertura de suelo desde junio del 2015 a la actualidad, actualizadas casi en tiempo real.

Desde allí se extrae casi toda la información del proyecto por lo que familiarizarse con las imágenes que entrega y las bandas que ellas poseen ayudara al entendimiento del script. Mas adelante se explica de forma rápida como se encuentra almacenada la información en las bandas de las imágenes.

En la siguiente página se encuentra más información:

https://developers.google.com/earth-engine/datasets/catalog/GOOGLE_DYNAMICWORLD_V1

REQUISITOS PREVIOS

El script se encuentra programado en R, por lo que junto a RStudio es necesario:

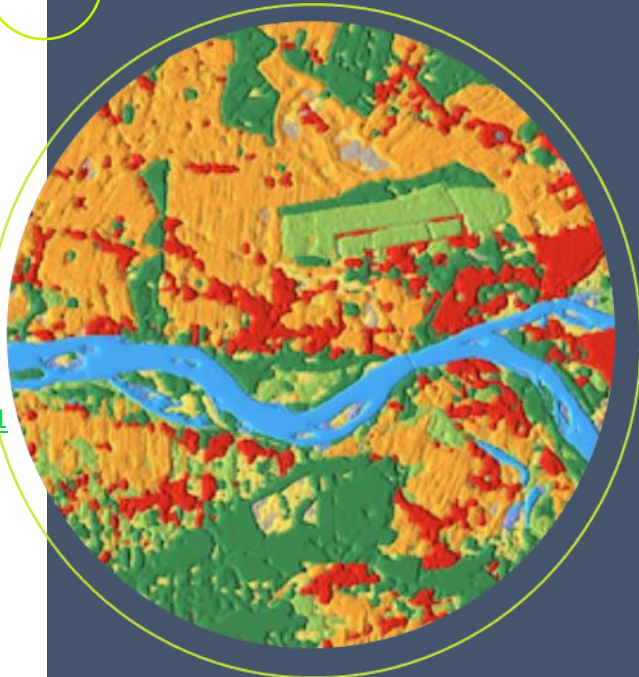
- Python versión 3 o superior
- Google Cloud SDK
- Cuenta capaz de acceder a Earth Engine Code
- Rtools

La librería principal del script es RGEE, esta se encarga de realizar la conexión con los servidores de Google Earth Engine, enviar y recibir las solicitudes.

RGEE utiliza Python para procesar las solicitudes. Además, para la conexión requiere de **Google Cloud SDK** instalado y también se requiere de una cuenta capaz de acceder a Earth Engine Code.

En caso de no tener una cuenta capaz de acceder a Earth Engine Code se puede solicitar una en el siguiente enlace:

https://earthengine.google.com/new_signup/



ESTRUCTURA DEL PROYECTO

El proyecto cuenta con 3 archivos:

0-land-coverage-using-earth-engine.R

Es el script principal encargado de cargar las librerías necesarias, llamar las funciones y mantiene la estructura general del proyecto.

1-data-download-and-format.R

Contiene las funciones necesarias para extraer datos desde Earth Engine y generar información.

2-data-display.R

Contiene las funciones necesarias para graficar la información generada.

PAQUETES

Los comandos para instalar los paquetes necesarios se encuentran en la sección de **Paquetes**, al inicio del script (Línea 9 a 16). Basta con instalarlos una vez.

- Remotes: instalar paquetes desde git
- Reticulate: interfaz a Python.
- GoogleDrive: manipular archivos en drive
- Rgee: earth engine API desde R.
- Terra: manipular datos espaciales, es decir los rasters.
- Tidyverse: colección de paquetes diseñados para manipular estructuras de datos.
- Stringi: parte de tidyverse.
- Plotly: crear gráficos.

Stringi se encuentra de forma separada ya que encontré un error durante la instalación de tidyverse y requirió de instalación manual.

PROBLEMAS CON RGEE

Durante el desarrollo de este proyecto y hasta el momento de escribir este manual rgee se encuentra aún inestable, esto debido a que hace un tiempo Google cambio la forma de acceder a su suite y han estado parchando rgee desde entonces.

En caso de encontrar algún error pueden buscar ayuda en el siguiente enlace:

<https://github.com/r-spatial/rgee/issues>

Normalmente para trabajar con rgee basta con ejecutar los comandos:

```
rgee::ee_install()
```

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

La primera vez que se intente iniciar sesión se deberá usar el navegador, pero luego se realizara de forma automática.

El error recurrente que nos encontramos fue que fallaba la ejecución de funciones de rgee al momento de reabrir el proyecto. Por ello, cada vez que se necesitaba trabajar se debía limpiar el entorno y reinstalar rgee. Como parche dejamos una lista de pasos a seguir para reinstalar rgee, esta se encuentra al comienzo de la sección de **Earth Engine** (Línea 22 a 36)

```
18 # ////////////////////////////////////// Earth Engine //////////////////////////////////////
19
20 # En caso de error seguir los pasos:
21 #
22 # Paso 1: Limpiar cache
23 # rm(list = ls(all.names = TRUE))
24 # gc()
25 #
26 # Paso 2: instalar rgee desde github
27 # remotes::install_github("r-spatial/rgee", force = TRUE)
28 #
29 # Paso 3: Reiniciar sesion y luego Limpiar entorno
30 # rgee::ee_clean_pyenv()
31 #
32 # Paso 4: Reiniciar sesion y continuar con la instalacion de ee
33
34 rgee::ee_install()
35
36 rgee::ee_initialize(drive = T)
37
```

ESTRUCTURA DEL SCRIPT

Una vez instalados los programas y paquetes que forman parte de los requisitos basta con ejecutar `rgee::ee_initialize(drive = T)` para iniciar sesión en la suite de Google y poder ejecutar las solicitudes a través de rgee.

En el script trabajamos con la provincia de cauquenes usando dos mapas, uno del año 2018 y otro del año 2021.

LIBRERIAS líneas 38-49

Primero es necesario cargar las librerías al entorno de trabajo de R para poder referenciar las funciones.

En caso de editar los archivos *1-data-download-and-format.R* o *2-data-display.R* es necesario volver a cargarlos al entorno de trabajo para reflejar los cambios.

FILTROS líneas 54-69

Segundo se deben crear los filtros para extraer la información de Dynamic World. Estos se componen de una parte espacial y otra temporal.

Para definir la parte espacial proponemos dos opciones:

1. La primera es usar zonas administrativas de primer nivel, es decir, regiones.
https://geonode.wfp.org/layers/geonode:adm1_gaul_2015
2. La segunda es usar zonas administrativas de segundo nivel, es decir, provincias.
https://geonode.wfp.org/layers/geonode:adm1_gaul_2015

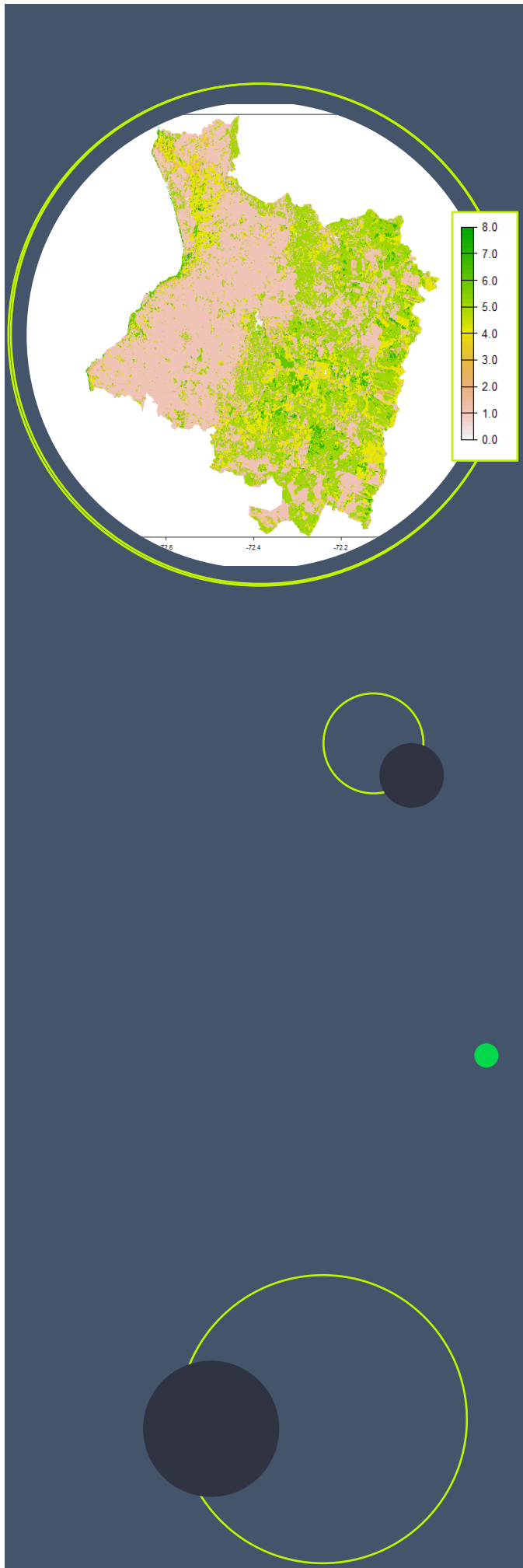
En los enlaces se pueden ver las zonas y la información disponible. Nosotros usamos los nombres administrativos para crear los filtros.

Las líneas de código de la 55 a la 58 se encargan de crear la geometría de la zona usando zonas administrativas de primer nivel. Las líneas de la 61 a la 63 se encargan de crearla usando zonas administrativas de segundo nivel. Solo una de ellas es necesaria, pero se pueden combinar ambos filtros espaciales en uno en caso de requerirse.

Para crear los **filtros** se combina la geometría con un rango de fechas. Nosotros usamos el mes de enero entero. Esto se puede ver en las líneas 68 y 69.

```
38 # ////////////////////////////////// Librerias //////////////////////////////////
39
40 library(remotes)
41 library(reticulate)
42 library(googleDrive)
43 library(rgee)
44 library(terra)
45 library(tidyverse)
46 library(plotly)
47
48 source("1-data-download-and-format.R")
49 source("2-data-display.R")
50
```





Ademas se definieron nombres para ambos mapas para facilitar el cambio de ellos en caso de ser necesario sin tener que editar todas las futuras referencias.

DESCARGAR IMÁGENES Y RASTERS *líneas 73-87*

Las imágenes de Dynamic world poseen 10 bandas: las primeras nueve (0 a 8) contienen la probabilidad de pertenecer a uno de los 9 tipos de suelo, la ultima banda representa el indice de la banda con la probabilidad mas alta.

La funcion `download_median_clipped()` Se encarga de obtener la mediana de las bandas de imágenes encontradas dentro del rango de fechas que se le entrega (*lineas 73-74*) . Esto da como resultado una unica imagen con sus respectivas 10 bandas que representa al grupo entero.

Esta imagen resultante es la que es entregada al Earth Engine para ser procesada y generar un raster mediante la funcion `download_raster()` (*linea 75-76*). Esta llamada se demora en ser procesada y actualmente es el cuello de botella en la velocidad del script.

El Raster generado por esta llamada no es optimo para trabajar, por esto se crea una copia en formato SpatRaster al momento de ser guardada al disco (*líneas 80-81*). Además, se cambia el formato desde números reales a enteros para habilitar la compatibilidad con Dinamica EGO.

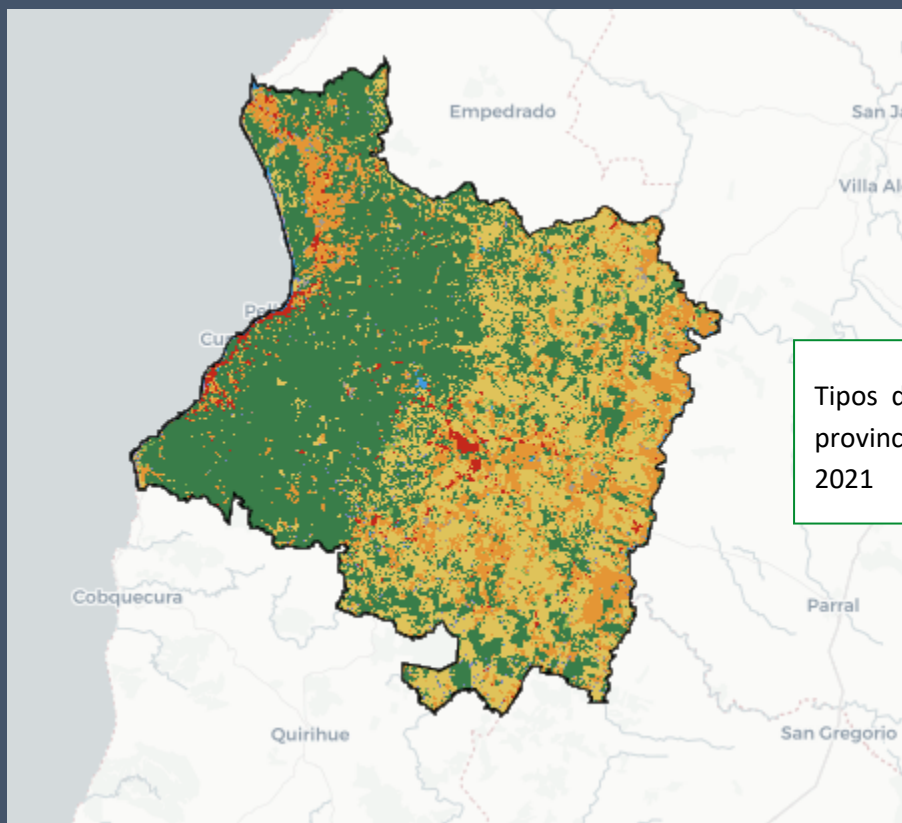
Finalmente se borra el Raster normal del disco y se carga en memoria el SpatRaster (*líneas 83-87*).

CREAR INFORMACION Y GRAFICAR *líneas 91-102*

A partir de los dos mapas se crean tablas de datos contando la cantidad de pixeles de cada cobertura de suelo y calculando cuanto porcentaje representa del total.

Con los datos de ambos mapas se calcula el cambio ocurrido en el intervalo para así graficar la información.

Finalmente se grafica el tipo de cobertura de ambos años usando la banda de imagen con el porcentaje mas alto. También se grafica la diferencia de cobertura entre ambos años. En la siguiente pagina se adjuntan las imágenes finales del script.



Tipos de cobertura en la provincia de Cauquenes, 2021

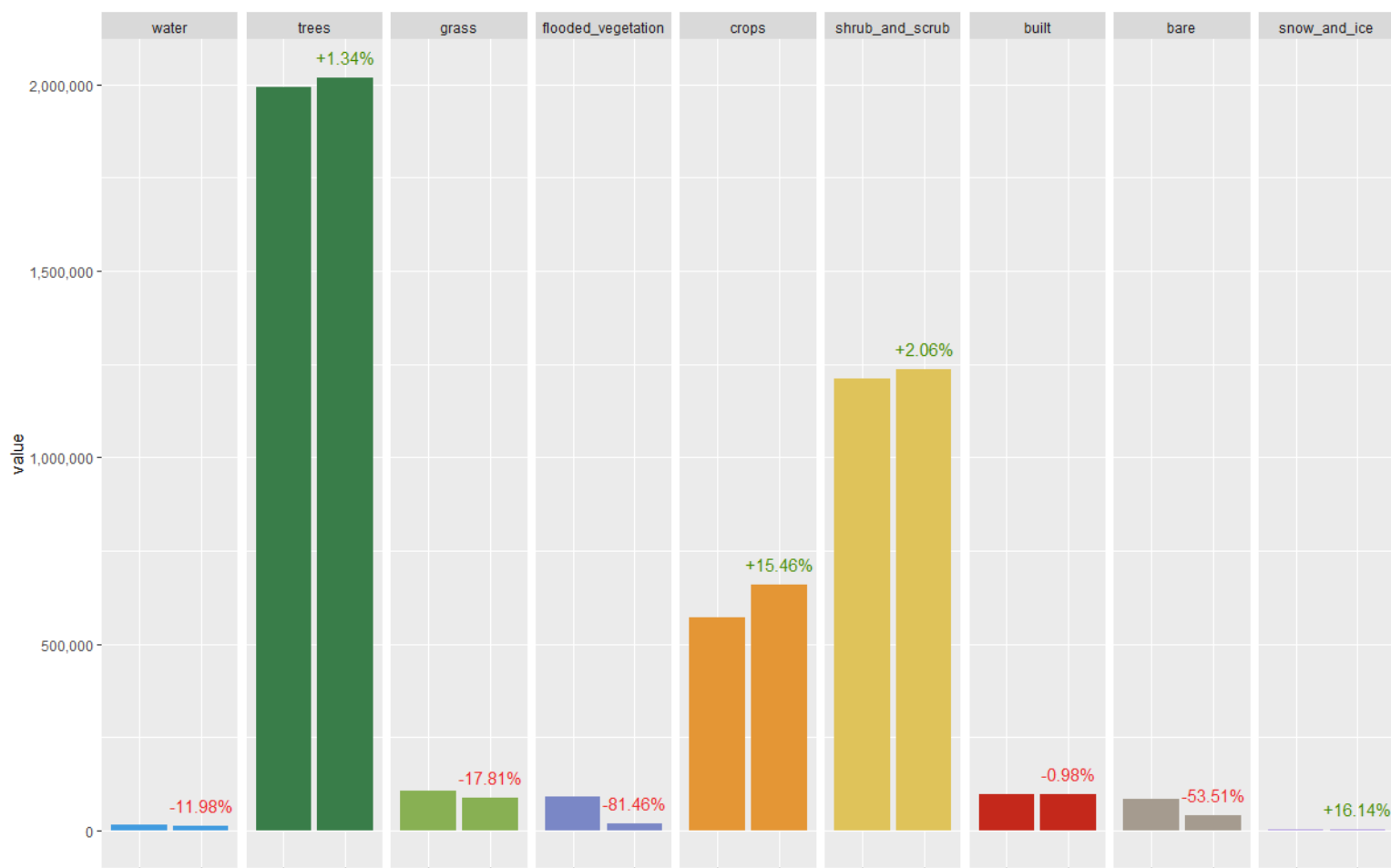


Gráfico de cambio en tipos de cobertura en la provincia de Cauquenes, 2021