Introduction to the HBRC package

Dan Richards

2023-11-23

Contents

Overview

This R package was developed as part of the LiDAR tools partnership programme between Hawke's Bay Regional Council and Manaaki Whenua - Landcare Research. The sub-project "Quantifying ecosystem services" was a seed / pilot project that aimed to investigate methods for using LiDAR data products in combination with other national- or global- scale datasets to estimate indicators of some ecosystem services.

The authors take no responsibility for the quality and accuracy of the models included in this R package. Ecosystem services modelling is highly uncertain, and the outputs of these models should not be solely relied on for decision-making.

Package installation and set-up

The hbrc package is now publicly available via the Manaaki Whenua Github page, in the hbrc repository. The package can be installed in a similar manner to another other Github-hosted package, directly from R using the `devtools` package. Installation instructions are available on the page.

This package has some dependencies that require additional installation - most are straightforward and mentioned in the documentation. The whitebox package for geoprocessing is best installed via Github, and requires some additional set-up that is well documented on the Github page.

Package functions summary

The `hbrc` package is designed for spatial modelling of ecosystem services using established methods. The core functionality for spatial analysis depends on the `terra` package for R, and all spatial inputs and outputs are in the terra formats (SpatRaster or SpatVector). The `hbrc` package includes functions for modelling indicators of ecosystem services. The following functions are provided;

Runoff retention

The `hbrc.runoff.cn` and `hbrc.runoff.e1` functions provide contrasting methods to estimate runoff retention by ecosystems.

`hbrc.runoff.cn` Follows an international Curve Number approach to estimate runoff retention given information on the land cover, underlying soil properties (Hydrological Soil Group) and incoming rainfall (Mockus, 1972; USDA, 1986). The curve number approach accounts for saturation and has been used in urban and rural studies in NZ (Richards et al 2023a; Richards et al. 2023b). The Curve Number method can also specify the antecedent soil moisture conditions (Mockus, 1972; USDA, 1986).

`hbrc.runoff.e1` Follows the NZ MBIE guidelines for the New Zealand Verification Methods and Acceptable Solutions Building Code Clause section E1. Surface Water (MBIE 2020). The E1 approach accounts for slope, land cover, and soil type, but not incoming rainfall.

Carbon stocks

The `hbrc.tree.c` function models aboveground biomass carbon stored in trees. This function estimates biomass carbon stored in each tree individually using an allometric approach, and therefore requires information on the spatial location of every tree, tree height, and (optionally) tree diameter at breast height. If

no diameter at breast height information is provided, it will be estimated from tree height using a published equation developed from tree records from Christchurch (Richards et al. 2023a, Quan et al., 2021).

The function provides a choice of two built-in allometric equations parameterised from studies in New Zealand; a general multi species allometric equation from Beets and colleagues (2012), and a multi-species urban tree allometric equation from Schwendenmann and Mitchell (2014).

Air pollution removal

The `hbrc.pm10` function models particulate matter (PM_{10}) removal from the air by tree and shrub leaves. This method follows an approach that has been developed and applied internationally (Escobedo and Nowak, 2009; Manes et al., 2014; Nowak et al., 1998). The method has also been applied in New Zealand (Cavanagh 2008; Cavanagh et al. 2009).

A key parameter in estimating air pollution removal is Leaf Area Index (LAI). LAI can be quantified spatially using several approaches, but here we use an empirical method based on Landsat imagery that was parameterised in Christchurch (Kato et al. 2013). This approach has previously been used to estimate air pollution removal in New Zealand (Richards et al 2023a). There is a separate `hbrc.lai` function for this part of the process.

Landscape aesthetics

The `hbrc.aesthetic` function models a relative indicator of landscape aesthetic attractiveness using a viewshed approach. The method is a composite of approaches for weighting the relative attractiveness of different landscape elements (Lavorel et al 2022; Byczek et al 2013), and a viewshed method that quantifies the visible area from different locations and weights visual importance by distance from the observer (Schirpke et al. 2013; Schirpke et al. 2016). A similar approach has been previously used in New Zealand (Richards et al. 2023b), although this function provides an improvement by weighting closer parts of the landscape as more important than visible parts that are further away (Schirpke et al. 2013; Schirpke et al. 2016).

Ultraviolet (UV) protection

The `hbrc.uv` function models UV protection by the tree canopy. This function uses an internationally-published method for estimating UV below a tree canopy and consequently the UV protection factor provided by trees (Na et al. 2014).

Shade

The `hbrc.shade` function models shade provided by the tree canopy. This function uses a rayshading model provided by the `rayshader` R package, and runs the model three times to quantify the total shade provided, and the contribution of vegetation in providing shade.

Nutrient retention

The `hbrc.nutrient` function models nutrient export by ecosystems. This function is based on the Natural Capital Project InVEST® Nutrient Delivery Ratio model which has been used around the world (Natural Capital Project, 2023). This model uses a simple mass balance approach to estimate the nutrient delivery ratio. Our version of the model looks only at the "surface" part of the model (which conceptually includes shallow subsurface flow). Please note that the model available in the `hbrc` package is not identical to the InVEST version, and so may give different results. Our model is also still in development, so please get in touch if you notice anything strange. If you would like to use the exact InVEST model via an R interface, you might be interested to try the rinvest package created by Jemma Stachelek.

Example usage

The following worked example shows the main functions applied to a test region where the relevant input datasets exist.

Test case study region

The case study region is in Hawke's Bay, just to the north of Wairoa. This is an area where detailed LiDAR dataset have been processed under the LiDAR partnership programme to extract terrain, canopy, and vegetation cover features. There are also other spatial datasets available that have been combined with LiDAR to generate new products. For example, the Land Cover DataBase (LCDB) has been extracted and fused with LiDAR to provide a detailed map of land cover. Let's load the available maps for this area and take a look in more detail;

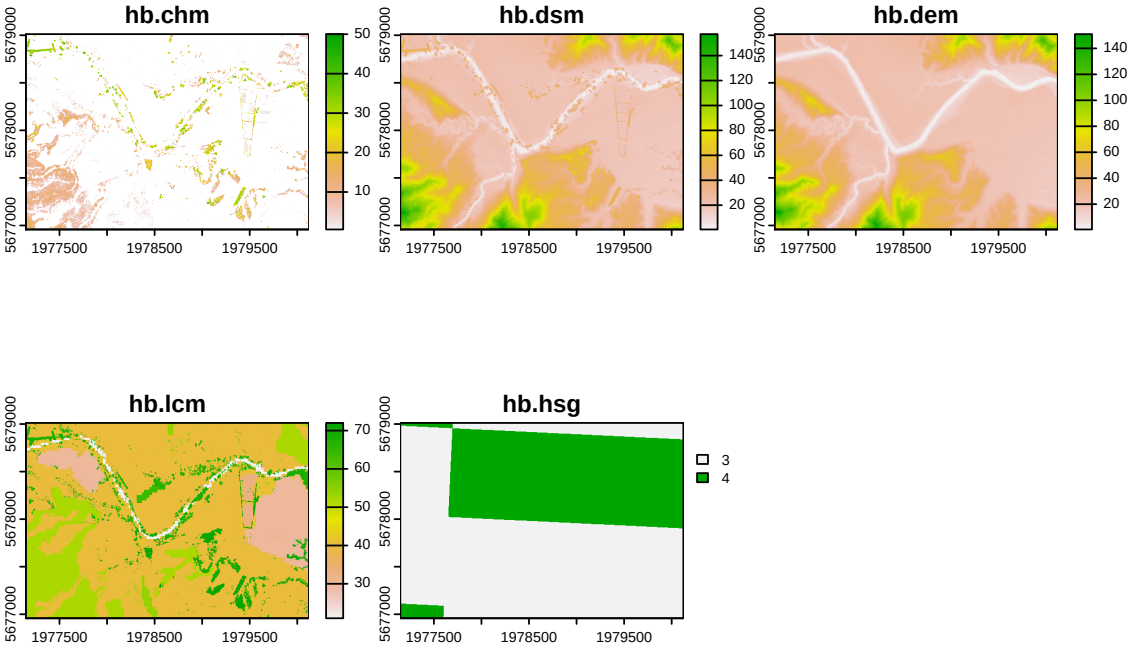
The maps for the case study area are included in the package. Due to a complication with the way that terra saves objects, the data are instead saved as objects for the older raster package. We will need to load the data files and then convert them to terra formats.

First let's load the relevant maps of land cover and 3D information, which are available at a ~1m resolution. The data are stored under the name "hb.1m" for this reason. The following code will read this data file and convert it to terra format.

```
# Read data
data("hb.1m")
# Convert to terra format
hb.1m <- as(hb.1m, "SpatRaster")
```

We can now take a look at the dataset. To quickly plot the maps, we can use plot().

```
# Plot maps
plot(hb.1m)
```



From this plot we can see that there are five data layers inside the "hb.1m" dataset. The first three refer to topographic data products that have been derived from the LiDAR point cloud. 1. "hb.chm" refers to a canopy height model for the area, which describes the height of vegetation above the ground in metres. 2. "hb.dsm" refers to a digital surface model, which describes the height of the surface of the point cloud in metres. 3. "hb.dem" refers to the digital elevation model, which describes topographic height in metres. 4. "hb.lcm" is a land cover map. This is represented as integer values, in which each integer corresponds to a different type of land cover. At the moment it is a mystery which types of land cover correspond to each integer value, but we shall get to that shortly. 5. "hb.hsg" is a map of soil Hydrologic Soil Groups (HSGs). This is represented as integer values, in which values 1:4 correspond to HSGs A:D, respectively.

You might notice some differences in how the data layers look. For example, the HSG map shows some large blocks, in contrast to the apparent higher resolution of the other layers. This is because it has been resampled from a global dataset (Ross et al. 2018).

Now let's study the land cover map in a bit more detail. This map has been produced by combining information from the LCDB with higher-resolution maps of tree cover obtained from LiDAR. The look-up table that describes the land cover types is available in the package as "eslookup". Let's load it and have a look at the top few rows.

```
# Load data
data("hb.eslookup")
# Look at the structure
dim(hb.eslookup)
```

```
## [1] 35 15
```

```
# Explore
head(hb.eslookup,3)
```

```
##   nid lcdb       name rc.hi rc.mi rc.li cn.a cn.b cn.c cn.d shade.veg uv.tree
```

```
## 1   1    0   Not land    NA    NA    NA   NA   NA   NA   NA          0       0
## 2   2    1   Built up  0.85  0.85  0.85   98   98   98   98          0       0
## 3   3    2 Urban Park  0.30  0.30  0.30   37   59   72   78          1       0
##   n.load n.retention l.attractiveness
## 1      0         0.0              0.0
## 2      0         0.0              0.0
## 3     10         0.7              0.5
```

It's quite a big table! There are 15 variables and 35 different land cover types. We can see that the second and third columns give the information that we were looking for - each value in column "lcdb" corresponds to a named land cover type in column "name". This table is going to be important as we go ahead with the ecosystem service modelling, because the other columns provide the look-up tables needed to run many of the models. Please note that the numbers given in this look-up table are estimates we have extracted from the literature. When running the models for your own study area, it would be valuable to read the papers that describe each model and parameterise your land cover values accordingly.

There are also a lot of rows, because this table is based on the original table provided by LCDB. Some of the vegetation cover types are not present in this table, for example we can find the coverage of mangrove by querying the hb.lcm map layer using the *global* function from terra.

```
# Sum of pixels
terra::global(hb.1m$hb.lcm==70,"sum",na.rm=TRUE)[1,1]
```

```
## [1] 0
```

It is unsurprising that there is no mangrove because the case study site is not close to the sea.

Runoff retention

To get a better understanding of how some of the functions work, we can now go ahead and try out the runoff retention function "hbrc.runoff.cn". This is the function for estimating runoff retention using a global curve number approach. You can use the help page to find out the required parameters for a function.

```
# Access help using ?
?hbrc.runoff.cn
```

This function needs a few items; 1. a land cover map lcm 2. a soil hydrologic groups map hsg 3. a lookup table of runoff curve numbers for each land cover type, for each of the four potential soil hydrologic groups 4. a value for precipitation in mm 5. a character describing the antecedent soil moisture conditions - either "dry", "average", or "wet" corresponding to the AMC classes I, II, and III respectively.

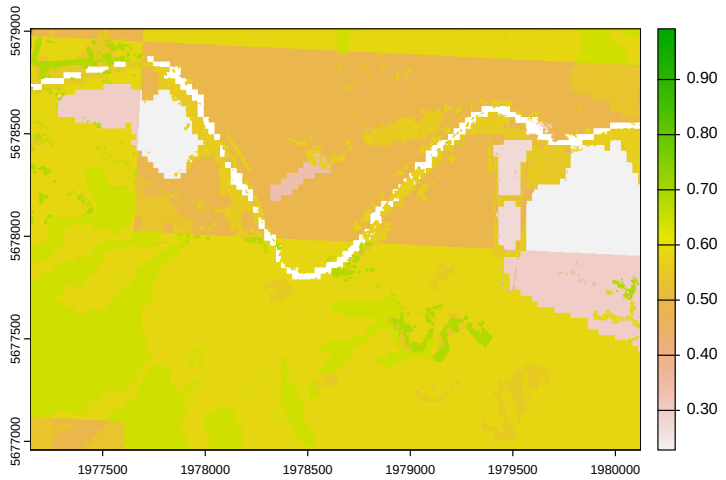
We have the first two items already, and the information for item 3 is included in our grand lookup table of ecosystem service parameter information (columns 7:10). We will assume a single precipitation value (90 mm) for the whole map, but we could equally use a raster instead, with different values in each pixel. We will also assume average antecedent conditions. Let's parameterise the function and run it, saving the output into a new object.

```
# Run hbrc.runoff.cn function
runoff.cn.out <- hbrc.runoff.cn(lcm = hb.1m$hb.lcm,
                                hsg = hb.1m$hb.hsg,
                                ltb = hb.eslookup[,c(2,7:10)],
                                precip = 90,
                                antecedent = "average")
str(runoff.cn.out)
```

```
## S4 class 'SpatRaster' [package "terra"]
```

As we can see from inspecting the resulting object, the function returns a single raster layer. The values in this layer are proportions representing the proportion of the total runoff retained. As we can see if we map the output, there are higher and lower areas, depending on the location of different land cover types and soils.

```
# Plot map
plot(runoff.cn.out)
```



There are some missing data (NA values) in the output map. If you'd like to see where these are, you can try `plot(is.na(runoff.cn.out))`. In the resulting map, you can see that the river area is marked as NA. This is because water bodies don't really act to retain runoff, as they immediately convey water. So, the look-up table that we have used excludes open water and some other habitats from providing runoff retention.

There is another way to model runoff retention provided in this package; the `hbrc.runoff.e1` function. You can use the help file for this function to learn more about this function, and will find that it needs some different parameters; 1. a land cover map lcm 2. a soil type map with three categories for high, mid, and low infiltration. 3. a map of slope 4. a lookup table of runoff curve numbers for each land cover type, for each of the three potential soil types

Since we don't have a soil type map, we can use the HSG map to infer it. A simple estimate would be to subtract a value of one from the HSG map, to give a three-category integer system `hb.1m$hb.hsg - 1`. We also don't have a slope map yet, but we can make one from the digital elevation model using the `terra::terrain()` function.

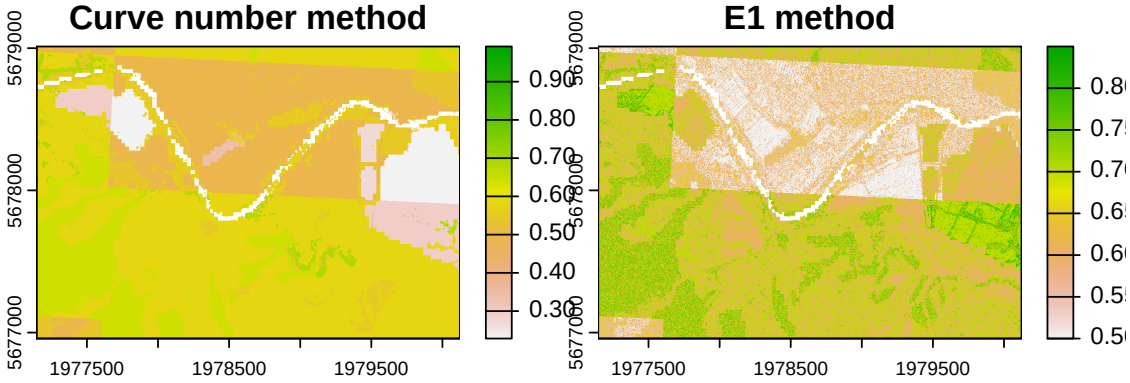
```
# Create slope map and convert from degrees to radians
hb.slope<- tan(terra::terrain(hb.1m$hb.dem, v="slope", unit="degrees"))*100

# Trim hb.slope
hb.slope[hb.slope>100]<-100
hb.slope[hb.slope<0]<-0

# Create soil type map
hb.stm <- hb.1m$hb.hsg - 1
hb.stm[hb.stm <1]<-1

# Run hbrc.runoff.e1 function
```

```
runoff.e1.out <- hbrc.runoff.e1(lcm = hb.1m$hb.lcm,
                                stm = hb.stm,
                                ltb = hb.eslookup[,c(2,4:6)],# Note different columns
                                slo = hb.slope)
str(runoff.e1.out)
```

```
## S4 class 'SpatRaster' [package "terra"]
```

Let's plot to compare

```
# Combine and plot
plot(c(runoff.cn.out, runoff.e1.out),
     main=c("Curve number method","E1 method"))
```



The results from both models are sensitive to the way that we assign parameters to each of the land covers. The curve number method will also give different results depending on how much precipitaiton there is, so feel free to experiment!

Aboveground biomass carbon stocks

For the sake of comparison, let's move on to a different type of ecosystem service model - one that doesn't need a land cover map at all. The spatial dataset required for the aboveground biomass carbon stock model is a SpatVector of tree locations, so we don't use any raster data to run this model. If you use `?hbrc.tree.c` you can open the help page for this model, and will find that it needs thre following information;

1. a SpatVector of tree locations `trees`

2. a vector of tree height values `treeheight`
3. a vector of tree diamter at breast height (DBH) values `treedbh`
4. an allometric equation to use `method`
5. a maximum tree height to trim trees to, which is used to make the estimates more conservative and robust to outlier records of unusually tall trees `maxheight`

We have provided a test dataset within the package. Again it is in the wrong format, but we can read it in and convert it.

```
# Read data
data("hb.trees")
# Convert to terra format
hb.trees <- as(hb.trees, "SpatVector")
# Inspect names
names(hb.trees)
```

```
## [1] "TUID"   "height" "radius"
```

The object `hb.trees` has been prepared from LiDAR data, and contains point locations for each tree. The three data columns provide a unique tree identification number, and tree-level information on canopy height and the radius of the canopy. Let's parameterise the function;

```
# Run function
treec.out<- hbrc.tree.c(hb.trees,
            treeheight = hb.trees$height,
            treedbh = NA,
            method = "beets",
            maxheight = 30)
names(treec.out)
```

```
## [1] "TUID"   "height" "radius" "kgc"
```

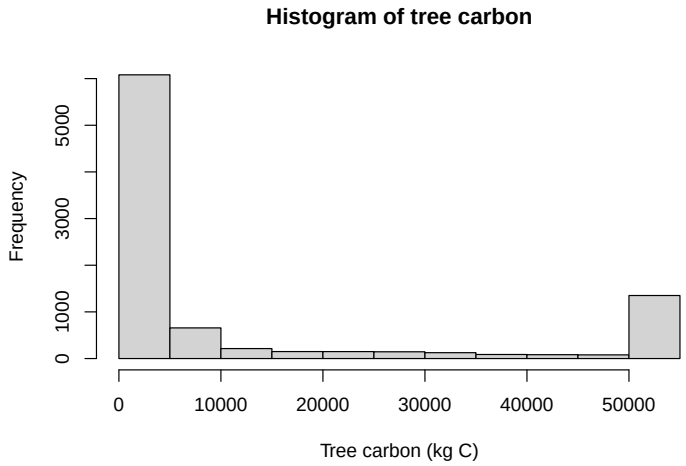
Note that we assigned the treedbh value to NA, since we don't have any data for that. This means that tree DBH values will be estimated from height using published equations. We also chose to thse the Beets et al. 2012 allometric equation to convert height and DBH into an estimate of carbon stocks. This is a general, multi-species equation that was developed empirically in New Zealand.

We can see the distribution of tree aboveground biomass carbon, and calculate the total within our study area, using the code below.

```
# Total stock, but convert to tonnes of C
sum(treec.out$kgc)/1000
```

```
## [1] 108472
```

```
# Frequency distribution
hist(treec.out$kgc, xlab = "Tree carbon (kg C)", main = "Histogram of tree carbon")
```



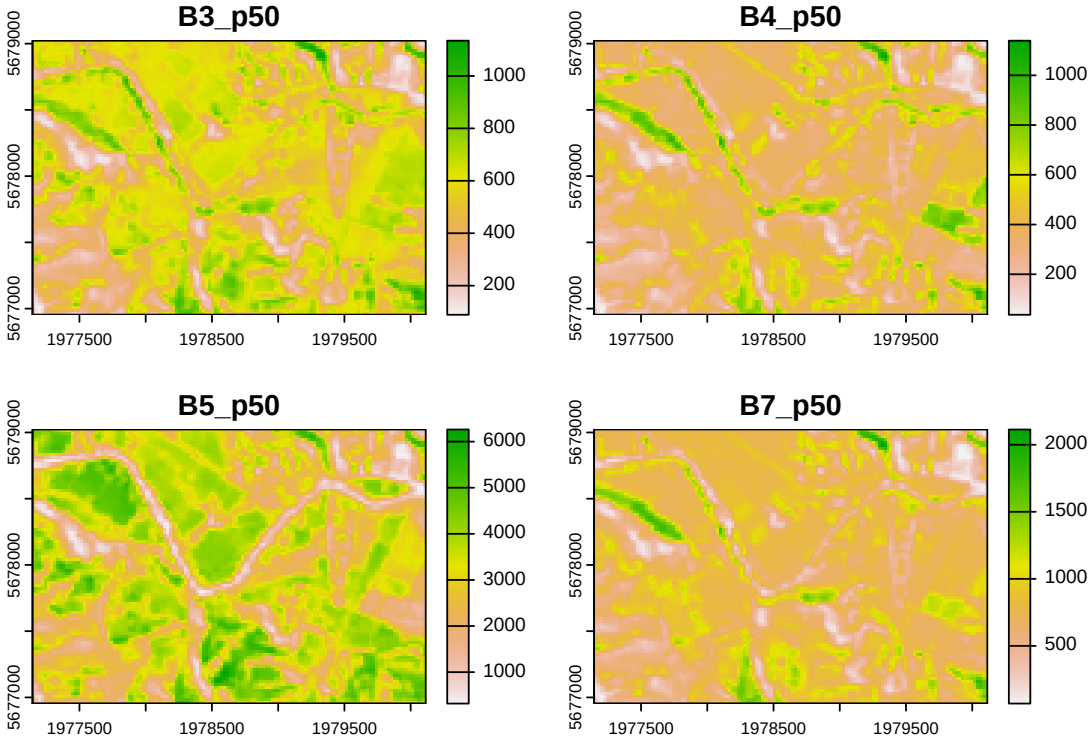
You can test how using a different equation might change the results - just use the "sm" method to use an equation developed for urban trees in Auckland.

Air pollution removal

So far we have used data derived from LiDAR alone (e.g. digital elevation mode) and a fusion of LiDAR and national land cover maps (e.g. the land cover map from LiDAR vegetation and LCDB). In the next model we will use satellite imagery extracted from the global Landsat archive.

It is possible to use individual Landsat images, but sometimes there can be patches of cloud, and they only represent one moment in time. Instead, we will use a cloud-free composite image that represents the median values extracted over a year. We have already done this analysis, and provide a test dataset within the package. Again it is in the wrong format, but we can read it in and convert it.

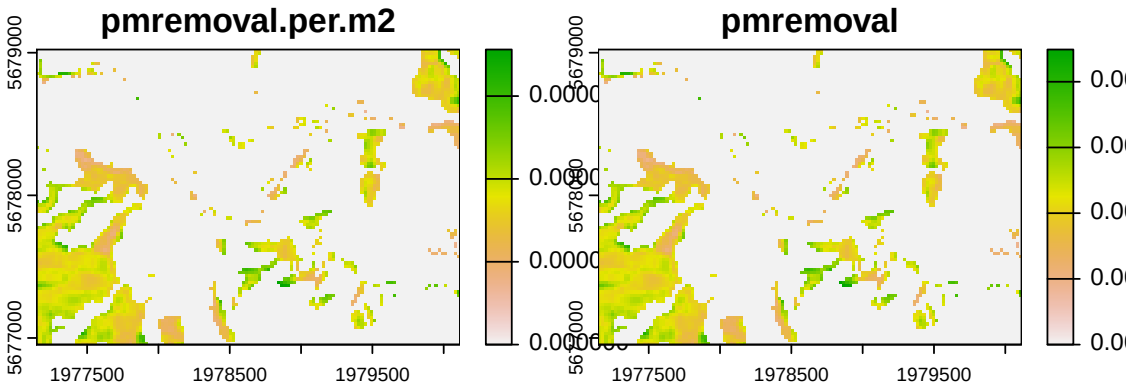
```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
# Read data
data("hb.lsat")
# Convert to terra format
hb.lsat <- as(hb.lsat, "SpatRaster")
# Plot the maps to inspect
plot(hb.lsat)
```



The plot shows that we have four layers, corresponding to Bands 3, 4, 5, and 7 from Landsat 8. This image will be used inside the `hbrc.lai` function to estimate leaf area index. We will then use the output map within the `hbrc.pm10` function in turn estimate the amount of particulate matter (PM_{10}) captured across the study area.

Since you are probably familiar with the way that this package works now, let's just parameterise the functions and take a look at the results. Feel free to inspect the help page using `hbrc.lai` or `?hbrc.pm10` to find more information.

```
# Run LAI estimation function
lai.out<- hbrc.lai(lcm = hb.1m$hb.lcm,
                   lsat = hb.lsat,
                   ltb = hb.eslookup[,c(2,12)])

# Parameterise PM10 function using the LAI map and other inputs
pm.out<- hbrc.pm10(lcm = hb.1m$hb.lcm,
                   lai = lai.out,
                   ltb = hb.eslookup[,c(2,12)])
plot(pm.out)
```



As you can see, we left some parameters unparameterised. In this case, the function will use the default values for background particulate matter concentration in the air `pm10` and deposition velocity `Vd`. While the parameterisation of deposition velocity is a commonly used number from the global literature, the background particulate matter concentration may be very different depending on your study area, and the time of the year. Data on particulate matter (PM_{10}) concentrations are available from a range of monitoring stations across New Zealand.

Ultraviolet (UV) protection

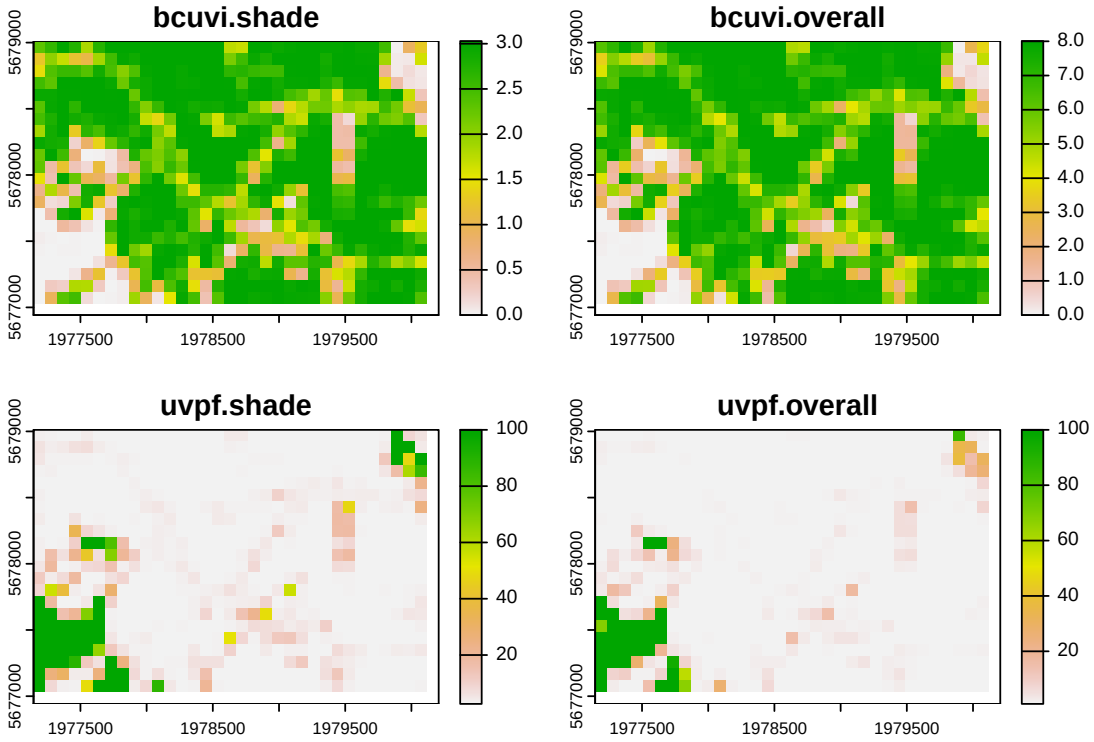
Exposure to UV is a major risk in Hawke's Bay. Tree canopies can provide some protection from UV, and this can be quantified - both in terms of an estimated exposure below the canopy, and also as a UV Protection Factor (UVPF) equivalent to the Sun Protection Factor value provided on bottles of sunscreen (Na et al 2014). The `hbrc.uv` function provides this model.

`hbrc.uv` requires some parameters that we have used before, and some others.

1. a land cover map `lcm`
2. a map of vegetation canopy height `chm`
3. a cutoff value for the minimum vegetation height to consider for providing shade `ccutoff`. We can set this at approximately head height (1.6 m) to ensure that shade is available to people.
4. a background UV index value `uvi`. UVI varies around the year, so we will apply a relatively high value but not extreme value of 8. It would also be possible to provide this parameter as a SpatRaster if there is spatial variation over the region of interest.
5. a solar zenith angle describing the sun position `sza`. Solar zenith angle depends on the time of day and year. We will assign a value of 25 degrees.
6. a rescaling factor for gridding the tree canopy cover dataset when calculating canopy cover `resc`. If we choose a rescaling factor of 10, our ~1 m raster will be resampled into ~10m chunks. Let's use a value of 100 m.
7. a lookup table that defines "tree" canopy cover `ltb`, similar to the particulate matter example.

8. a sky condition values, either as a single value or SpatRaster. This can have multiple values which are listed in `data(skylookup)`. For simplicity we will assume that the sky condition is clear ("CLR").

```
# Run function
uv.out<- hbrc.uv(lcm=hb.1m$hb.lcm,
                 chm=hb.1m$hb.chm,
                 ccutoff=1.6,
                 uvi=8,
                 sza=25,
                 resc=100,
                 ltb = hb.eslookup[,c(2,12)],
                skc="CLR")
plot(uv.out)
```



You can see from the resulting maps that the function provides four layers of output, which describe different variables. The top two plots provide the below-canopy modelled UVI. `bcuvi.shade` gives the UVI modelled in areas shaded by trees in each grid cell of the map, while `bcuvi.overall` gives us the average UVI (thus accounting for the relative area of tree-covered and shadeless areas) in each grid cell. The bottom two maps give the UVPF (UV protection factor) for the below canopy shade and average conditions. UVPF is an indicator equivalent to how many times longer a person would have to spend in a particular environment to receive the same exposure as in an open location with no solar UV protection (Na et al 2014). The function has an in-built constraint such that the UVPF value cannot be greater than 100.

Shade

So far we have used models that are based on literature values, empirical relationships, and general patterns. A different approach to modelling ecosystem services is to use a mechanistic model. Mechanistic models can be useful for services that have a physical basis, such as shade. Shade modelling can be achieved using a

mechanistic or "rayshading" model of light througha 3-dimensional virtual environment, and one benefit of LiDAR data is that they can provide a detailed 3D representation of the world.

The `hbrc.shade` model provides a function to run mechanistic rayshading models that allow us to estimate the shade provided under given light conditions, and thus quantify the relative contributionof trees in providing shade (as opposed to shade provided by topography or buildings).

The code below provides an example of running the `hbrc.shade` model for our study area. Most of the required parameters we have come across before, or you can read about in the help page. There are a couple that need some explanation;

- `resc` in this function refers to a rescaling factor, which should be given as an integer. Rescaling can be valuable when running this mechanistic model, because the simulation of light through a 3D environment is reasonably computationally intensive. We are only dealing with a small case study area, but even so we will rescale by a factor of 10 to speed up the computation time.
- `sv` is a matrix of sun azimuths and zeniths. The number of rows corresponds to the number of time points that you would like to simulate. This matrix can be produced from a vector of dates and time, location on earth, and timezone, using the `insol` package. An example is provided in the help file (and repeated below). The results you get from the shade model will be influenced by the dates and times that you use to run the model, so take note of that. More information on specification of the sun azimuth and zenith function can be found within the `insol` package documentation.

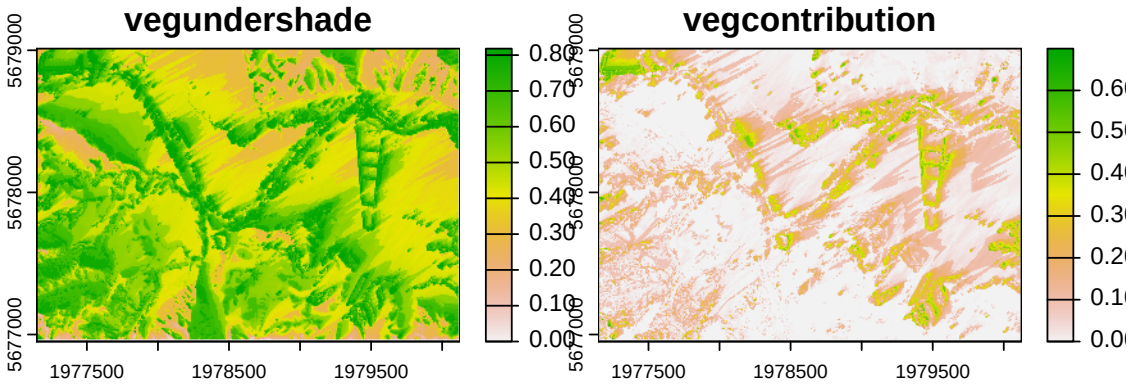
```
# Create sv
svv<- insol::sunpos(insol::sunvector(
 c(insol::JD(seq(ISOdate(2019,02,01,8),ISOdate(2019,02,01,18),by='hour'))),
 latitude = -43.52,
 longitude = 172.64,
 timezone=13))

# Run function
shade.out<- hbrc.shade(lcm=hb.1m$hb.lcm,
                   chm=hb.1m$hb.chm,
                   ccutoff=1.6,
                   resc=10,
                   dsm = hb.1m$hb.dsm,
                   sv= svv)
names(shade.out)
```

```
## [1] "vegshade"        "novegshade"      "vegundershade"   "vegcontribution"
```

You may notice that this function takes a little longer to run than some of the earlier ones. Be careful (or prepared to wait) when modelling a large area, and think carefully about the best way to use the `resc` parameter to balance the trade-off between spatial precisions and computation time.

The output of `hbrc.shade` is a RasterStack with four layers, which you can explore or find more information about using the help page. Let's plot the proportion of time that shade is provided by and under vegetation (`vegundershade`) as well as the proportion of the shade provided by vegetation (`vegcontribution`).

```
# Plot required maps
plot(shade.out[[3:4]])
```



We can see that shade is provided more than 80% of the time in many places, and that vegetation provides more than 60% of the total shade. You might also notice that the results of this model provide higher-resolution maps than we obtained from the previous `hbrc.uv` model (which is also somewhat about "shade"). The `hbrc.shade` model can achieve high resolution outputs partly because of the mechanistic model approach. While these models both relate to shade prvoided by vegetation, they give us different indicators of the ecosystem service provided. Shade is a useful indicator for many purposes that are not connected to people (e.g. for understanding shade cover in waterways), while from a human health perspective UV protection might be a more pertinent and informative indicator. It all depends on the question that we are interested in.

Landscape aesthetics

We have previously considered only regulating ecosystem services, but nature also provides critical contributions to recreation, cultural heritage, and psychological well-being. The `hbrc.aesthetic` function provides a means to score the relative attractiveness of different parts of a landscape, using a combination of a mechanistic method to model the physical area of the landscape that can be viewed from specfiic locations, and a look-up table of relative attractiveness scores for different land cover types. The function needs some parameters

1. a land cover map `lcm`
2. a vegetation canopy height model `vhm`
3. a digital elevation model `dem`
4. a value for the height of the observer `observerh` which we can fix to head height (1.6 m).
5. the number of locations across the landscape to analyse `subs`. This value will greatly impact the detail we see in the output maps, but also the computation time. Because we need to analyse (in 3D) the landscape viewshed for each location, this function is quite computationally intensive.
6. a lookup table `ltb` that includes relative attractiveness of different land cover types. We can use the values in `hb.eslookup$l.attractiveness`.

7. a rescaling factor `resc` which can be used to speed up the computation time. We will not provide this value, so the function will use the default which is `NA` meaning no rescaling will be applied.

Let's run the function for 30 viewing points across the area.

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
# Silence terra progress bar
terra::terraOptions(progress=0)
# Run function
aesthetic.out<- hbrc.aesthetic(lcm=hb.1m$hb.lcm,
                    chm=hb.1m$hb.chm,
                    dsm = hb.1m$hb.dsm,
                    observerh=1.6,
                    subs=30,
                    ltb = hb.eslookup[,c(2,15)])

# Inspect names
names(aesthetic.out)
```

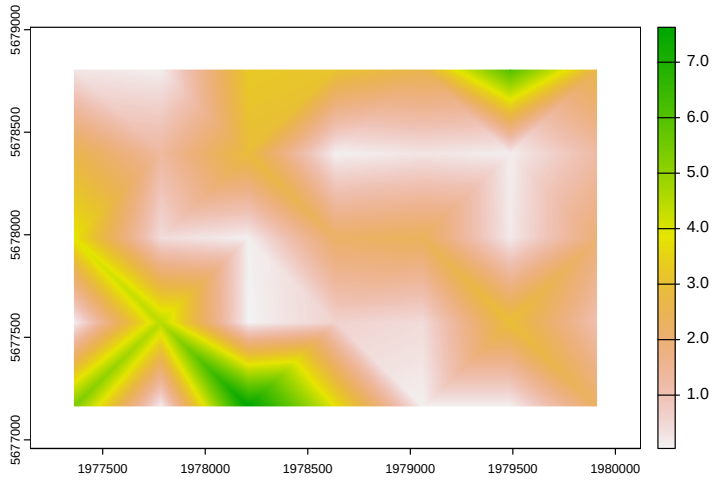
```
## [1] "lascore"       "vsarea"        "combinedscore"
```

The output of this function is a SpatRaster showing the interpolated aesthetic value scores from the locations (three layers of data). The three layers in the SpatRaster show different indicators;

1. `lascore` which is the aesthetic score viewed from each point.
2. `vsarea` which is the viewshed area visible from each part of the landscape (a larger value indicating a bigger area can be seen).
3. `combinedscore` a combination of these two scores defined by multiplying them against each other.

Let's plot only the combined score for now.

```
# Plot
plot(aesthetic.out[[3]])
```



We can see that the resolution is very low. This is because the function was parameterised to model only 30 locations spaced in a regular grid, and then linearly interpolate between them. If you have time, you could try running the function again with a larger number of subsampled locations to create a more detailed map.

Nutrient retention

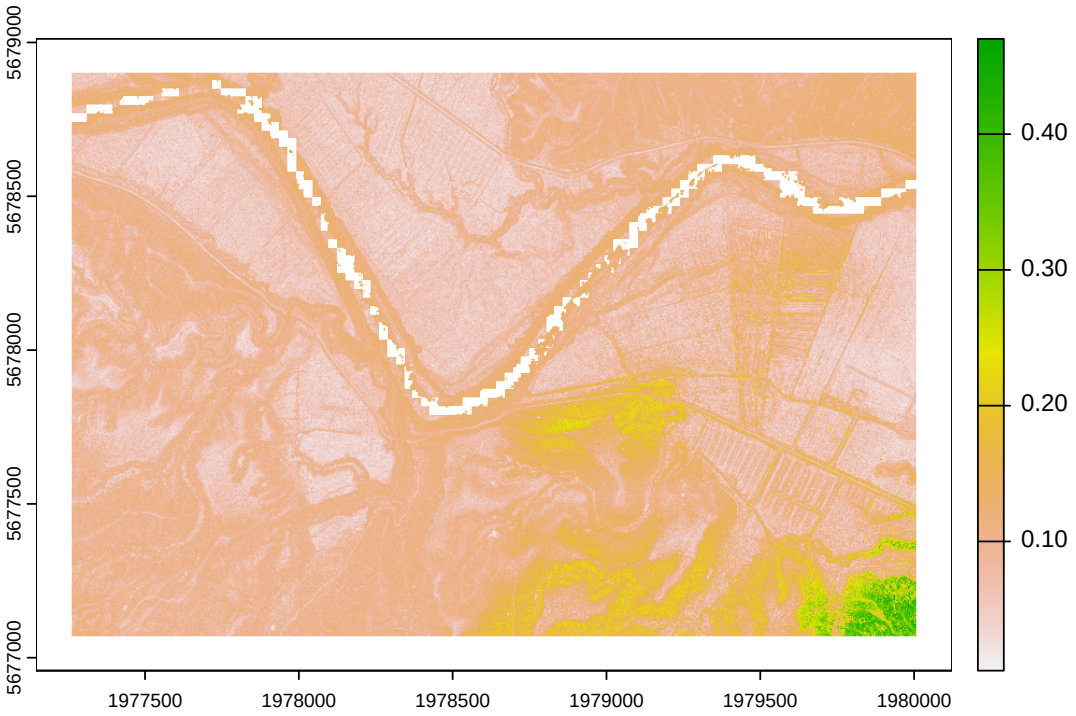
The `hbrc.nutrient` function requires many parameters that we are familiar with. It also requires a runoff index map, which we can create from the outputs of the `hbrc.runoff.cn` function by multiplying the proportion of runoff by the rainfall. Similar to the landscape aesthetic model above, we cannot run the model across the whole area, so need to select a number of subset points.

The `hbrc.nutrient` function requires a look up table with information about the nutrient loads and retention efficiency of each land use. An example of generating the table is provided in the help file, and included below. The function also requires a look up table indicating water, which we create in the code below.

Finally, the function has two parameters that influence the computational time.We can alter the resolution via the rescaling factor `resc`, and alter the number of subsampled points that are modelled and interpolated, using the subsampling factor `subs`. We have come across both of these before, in the shade and landscape aesthetic models respectively. The largest computational influence is the `subs` parameter, for which we will generate a grid of 120 points. We might also want to use `resc`. Using a simple mass balance approach, a resolution of 1 metre by 1 metre pixels is probably more detailed than the precision of the model, so let's rescale it by a factor of 10.

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
# Silence terra progress bar
terra::terraOptions(progress=0)

# Prepare runoff index
runoff.index<- (1-runoff.cn.out)*90

# Prepare BPT variable (look up table)
bpt<-data.frame(
 LULC_desc = hb.eslookup$name,
 lucode = hb.eslookup$lcdb,
 load_n= hb.eslookup$n.load,
 eff_n= hb.eslookup$n.retention,
 crit_len_n= 25)

# Prepare ltb with a binary look up table for which areas are water
ltb<-cbind(hb.eslookup$lcdb,
           0)
 ltb[11:13,2]<-1

# Run function
nutrient.out<- hbrc.nutrient(lcm=hb.1m$hb.lcm,
                  dem=hb.1m$hb.dem,
                  runoff=runoff.index,
                   subs=120,
                  resc=NA,
                   bpt =bpt,
                  ltb=ltb)
```

```
# Plot
plot(nutrient.out)
```



The output of the `hbrc.nutrient` function is a raster map of the proportion of the nutrient input per pixel that is exported from the pixel. The InVEST documentation makes it clear that the model does not "directly quantify the amount of nutrient retained on the landscape",and is better suited for making comparisons between scenarios in terms of the provision of nutrient retention ecosystem services.

Uncertainty propagation

Until now, we have applied all of the models using fixed parameters, thus implicitly assuming that we know these parameters exactly. In reality the world is very uncertain, and we rarely have a completely confident and precise knowledge of all of the parameters that are required. It is important to be aware of these uncertainties when using the results of ecosystem service models. One approach for quantifying the impact that parameter uncertainty has on modelled outcomes is through a kind of sensitivity analysis where we propagate uncertainty in our knowledge of the parameters through the model and quantify the variability in outcomes (Richards et al . 2020).

Let's look at the air pollution removal model as an example, because it is quite simple and quick to run. We will assume that uncertainty in this model comes from two places. First, our maps of leaf area index are uncertain due to variation in the empirical equation used to infer LAI from the Landsat satellite date. In the background of the `hbrc.lai` model is an equation developed empirically from data published in Kato et al 2013. The sigma value for that regression equation is 0.588, which we can use to draw values randomly from a normal distribution to represent variation around the fitted regression estimate. In R code this is achieved through the `rnorm` function.

We also need to propagate uncertainty associated with the parameterisation of the deposition velocity `Vd`. The standard approach is to use a deposition velocity accounting for resuspension of 0.0064 m/s (Nowak

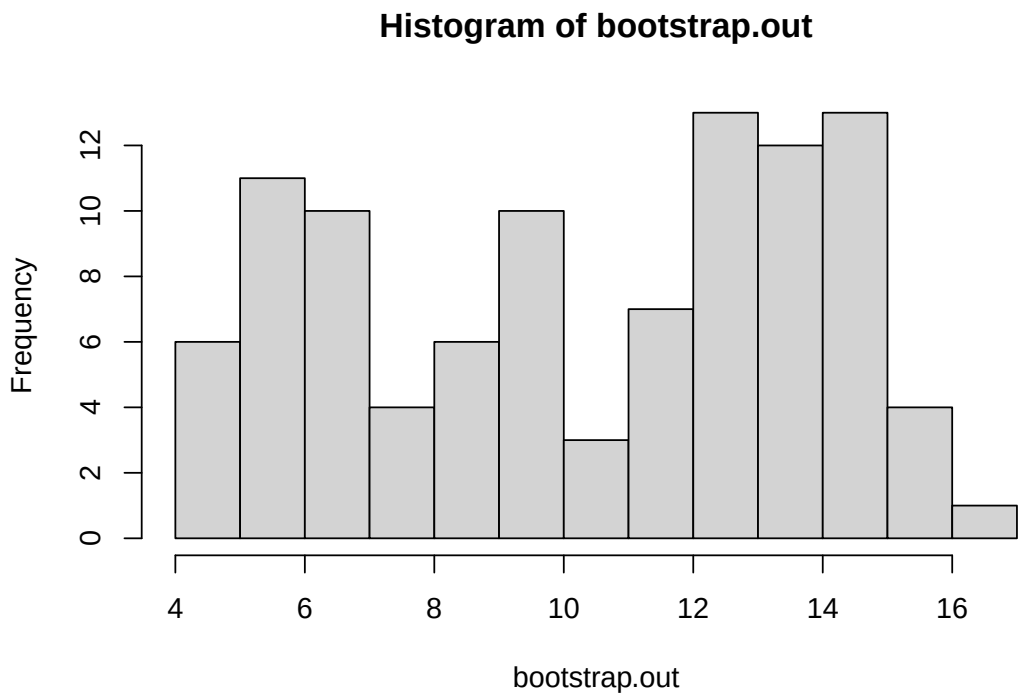
et al. 2008). According to Nowak and colleagues, this value is a median Vd for particles taken from an earlier review article (Lovett et al. 1994) which is a great paper that highlights uncertainty at many times. The Lovett article gives a range of possible values for large particle deposition velocity, equivalent to values of between 0.0025 and 0.01 m/s. So for our uncertainty propagation, we will assume that `Vd` can range uniformly between these values, using the random uniform distribution or `runif` function.

In the code below we propagate the uncertainty involved in estimating `LAI` and assigning a `Vd` parameter using a permutation or bootstrap method. We run a simulation 100 times, and each time we randomly modify our `LAI` and `Vd` parameters. After running the `hbrc.pm10` for each bootstrap permutation, we extract the total mass of PM_{10} removed by tree leaves across the landscape.

```
# Prepare empty vector to save outputs
bootstrap.out<-rep(NA, 100)

# Start loop for the n permutations
for(i in 1:length(bootstrap.out)){

  # Create temporary LAI raster dataset
  lai.boot<- lai.out

  # Draw LAI modification value from normal distribution and add to lai.out
lai.boot <- app(lai.out, rnorm, sd = 0.588) + lai.boot

# Remove non tree values
lai.boot[lai.out==0]<-0

# Remove 0 LAI because it is impossible
lai.boot[lai.boot<0]<-0

# Run model
pm.boot<- hbrc.pm10(hb.lcm,
                    lai.boot,
                    Vd = runif(1, min= 0.0025, max =0.01),
                    ltb = hb.eslookup[,c(2,12)])
bootstrap.out[i]<- global(pm.boot[[2]], "sum",na.rm=TRUE)[1,1]

}
```

The bootstrap can take some time to run, but can be made less or more computationally expensive by varying the number of permutations. We can plot a frequency distribution of the outcomes to look at the uncertainty in our estimate. We can also take the quantiles of the outcomes that we saved in `bootstrap.out` to generate a bootstrap confidence interval for our total estimate. By convention, we use the 2.5 and 97.5 percentiles to give the 95% confidence interval. The code below plots the frequency distribution and calculates these confidence intervals as well as the median estimate.

```
# Plot frequency distribution
hist(bootstrap.out, breaks = 10)
```




```
# Confidence interval
quantile(bootstrap.out, c(0.025, 0.5,0.975))
```

```
##      2.5%       50%     97.5%
##  4.259938 10.960059 15.407131
```

So there we go. Our estimate of total PM_{10} removed on one day in this section of the landscape, assuming a background concentration of 22 g/ m3, would be ~9.3 kg. The 95% confidence intervals around this estimate ranged between 4.7 kg and 14.9 kg, so quite a big uncertainty! Please note that your numbers might be slightly different, such is the nature of the method.

This was a quick example to highlight approaches for quantifying uncertainty, hopefully it has been helpful in explaining the importance of thinking about model uncertainty when quantifying ecosystem services over space.

Saving output maps

To use the results of the ecosystem service models in other applications, we can use the normal functions provided by the `terra` package for writing files. For example, to save the output of the nutrient retention model, we can use the `writeRaster` function which requires the SpatRaster object we want to save, and the filepath for where we want to save it.

```
# Save nutrient.out
writeRaster(nutrient.out, "nutrient.tif", overwrite=TRUE)
```

That's it! All saved to the working directory (it would be much better to specify a real file path)...

Acknowledgements

The work was supported by: the Strategic Science Investment Funding for Crown Research Institutes from the New Zealand Ministry of Business, Innovation and Employment's Science and Innovation Group. This work was also supported by the Environmental Science section of Hawke's Bay Regional Council via the LiDAR tools partnership project Contract Number HBRC-22-716.

The test datasets included in this package were provided by Jan Schindler. The methods for preparing these data were developed with support from the MBIE Catalyst: Strategic Fund through the programme "Bridging the gap between remote sensing and tree modelling" Contract C09X1923.

Citation

Most of the ecosystem services models used in the package can be cited via the original references given in the documentation, and collectively via our study applying ecosystem service models to Mackenzie District;

Richards DR, Herzig A, Abbott A, Ausseil A-G, Guo J, Sood A, Lavorel S (2023b). Diverse contributions of nature to climate change adaptation in an upland landscape. Ecosystems and People 19, 2225647.

If you'd like to cite the package itself, something like this would be fine;

Richards DR (2023). hbrc: Functions for estimating ecosystem services indicators using high-resolution spatial datasets. https://github.com/manaakiwhenua/hbrc

References

Byczek, C., Longaretti, P.Y., Renaud, J., Lavorel, S., 2018. Benefits of crowd-sourced GPS information for modelling the recreation ecosystem service. PLoS One 13. https://doi.org/10.1371/journal.pone.0202645

Cavanagh, J., 2008. Influence of urban trees on air quality in Christchurch; preliminary estimates. Landcare Research Report LC0708/097.

Cavanagh, J.A.E., Zawar-Reza, P., Wilson, J.G., 2009. Spatial attenuation of ambient particulate matter air pollution within an urbanised native forest patch. Urban Forestry and Urban Greening. 8, 21–30. https://doi.org/10.1016/j.ufug.2008.10.002

Escobedo, F.J., & Nowak, D. J. (2009). Spatial heterogeneity and air pollution removal by an urban forest. Landscape and Urban Planning, 90(3-4), 102-110. https://doi.org/10.1016/j.landurbplan.2008.10.021

Kato, A., Morgenroth, J., Kelbe, D., Gomez, C., Van Aardt, J., 2013. Ground truth measurement of trees using terrestrial laser for satellite remote sensing. International Geoscience Remote Sensing Symposium. 2106–2109. https://doi.org/10.1109/IGARSS.2013.6723228

Lavorel, S., Grigulis, K., Richards, D.R., Etherington, T.R., Law, R.M., Herzig, A., 2022. Templates for multifunctional landscape design. Landsc Ecol 37, 913–934. https://doi.org/10.1007/s10980-021-01377-6

Lovett, G. M. (1994). Atmospheric deposition of nutrients and pollutants in North America: an ecological perspective. Ecological Applications, 4(4), 629-650.

Manes, F., Silli, V., Salvatori, E., Incerti, G., Galante, G., Fusaro, L., & Perrino, C. (2014). Urban ecosystem services: Tree diversity and stability of PM10 removal in the metropolitan area of Rome. Annali Di Botanica, 4, 19–26. https://doi.org/10.4462/annbotrm-11746

MBIE (2020). Acceptable solutions and verification methods for New Zealand building code clause E1. Surface Water. First Edition, Amendment 11. https://www.building.govt.nz/building-code-compliance/e-moisture/e1-surface-water/

Mockus V (1972). Hydrology, in: Soil Conservation Service National Engineering Handbook. US Soil Conservation Na HR, Heisler GM, Nowak DJ, Grant RH (2014). Modeling of urban trees' effects on reducing human exposure to UV radiation in Seoul, Korea. Urban Forestry & Urban Greening, 13(4), 785-792.

Natural Capital Project, 2023. InVEST 3.14.0. Stanford University, University of Minnesota, Chinese Academy of Sciences, The Nature Conservancy, World Wildlife Fund, Stockholm Resilience Centre and the Royal Swedish Academy of Sciences. https://naturalcapitalproject.stanford.edu/software/invest

Nowak, D. J., McHale, P. J., Ibarra, M., Crane, D., Stevens, J. C., & Luley, C. J. (1998). Modelling the effects of urban vegetation on air pollution. In S. Grynin & N. Chaumerliac (Eds.), Air Pollution Modelling and Its Application XII (pp. 399–407). Plenum Press.Service, Washington, DC, USA, p. 762.

Quan W, Sullivan JJ, Meurk CD, Stewart GH, 2021. A taxonomically detailed and large-scale view of the factors affecting the distribution and abundance of tree species planted in private gardens of Christchurch city, New Zealand. PeerJ 9.https://doi.org/10.7717/peerj.10588

Richards DR, Polyakov M, Brandt A, Cavanagh J, Diprose G, Milner G, Ramana JR, Simcock R (2023a). Inequity in nature's contributions to people in Ōtautahi/ Christchurch: A low-density post-earthquake city. Urban Forestry and Urban Greening 86, 128044.

Richards DR, Herzig A, Abbott A, Ausseil A-G, Guo J, Sood A, Lavorel S (2023b). Diverse contributions of nature to climate change adaptation in an upland landscape. Ecosystems and People 19, 2225647.

Richards DR, Tan BAT, Gaw LYF, Masoudi M (2021). NCS2020: An R package for modelling urban ecosystem service provision in Singapore. https://github.com/drexrichards/nzes

Richards DR, Thompson BS, Wijedasa L (2020). Quantifying net loss of global mangrove carbon stocks from 20 years of land cover change. Nature Communications 11: 4260.

Ross, C.W., L. Prihodko, J.Y. Anchang, S.S. Kumar, W. Ji, and N.P. Hanan. 2018. Global Hydrologic Soil Groups (HYSOGs250m) for Curve Number-Based Runoff Modeling. ORNL DAAC, Oak Ridge, Tennessee, USA. https://doi.org/10.3334/ORNLDAAC/1566

Schirpke, U., Tasser, E., Tappeiner, U., 2013. Predicting scenic beauty of mountain regions. Landsc Urban Plan 111, 1–12. https://doi.org/10.1016/j.landurbplan.2012.11.010

Schirpke, U., Timmermann, F., Tappeiner, U., Tasser, E., 2016. Cultural ecosystem services of mountain regions: Modelling the aesthetic value. Ecol Indic 69, 78–90. https://doi.org/10.1016/j.ecolind.2016.04.001

Tan BA, Gaw LYF, Masoudi M, Richards DR (2021). Nature-based solutions for urban sustainability: An ecosystem services assessment of plans for Singapore's first "forest town". Frontiers in Environmental Science 9: 610155.

USDA (1986). Urban Hydrology for Small Watersheds. In United States Department of Agriculture (Issue Technical Release 55 (TR-55)).