Comp372
16.08.2021

GHG APP DEVELOPERS MANUAL

THE UNIVERSITY OF OTAGO

MONIQUE ARRON & BRETT JONES

Developer's manual documenting the system architecture, data & methods.

Abstract

The University of Otago has set a target to achieve net zero greenhouse gas emissions by 2030. We have built an interactive web application that allows users to view the university's current greenhouse gas emissions forecast. The Greenhouse Gas Emissions Forecasting app features three sliders for the user to interact with. Enabling them to view the impacts that growth/decline of student numbers, level of behaviour change and the pace at which New Zealand's electricity grid transitions to 100% renewable energy. Based on the total emissions forecast on a reactive graph, the number of hectares to plant and the cost to purchase carbon credits to offset these emissions, will update. This developers manual includes explanation and instructions for updating the app.

Table of Contents

Introduction

This developers manual has been compiled for developers at the University of Otago, to enable them to update and deploy the University of Otago Greenhouse Gas Emissions interactive app. The purpose of this app is to aid the client Craig Cliff (programme manager for Net Carbon Zero at the University of Otago’s Sustainability Office), in facilitating awareness and informed decision-making among university stakeholders and any other interested parties.

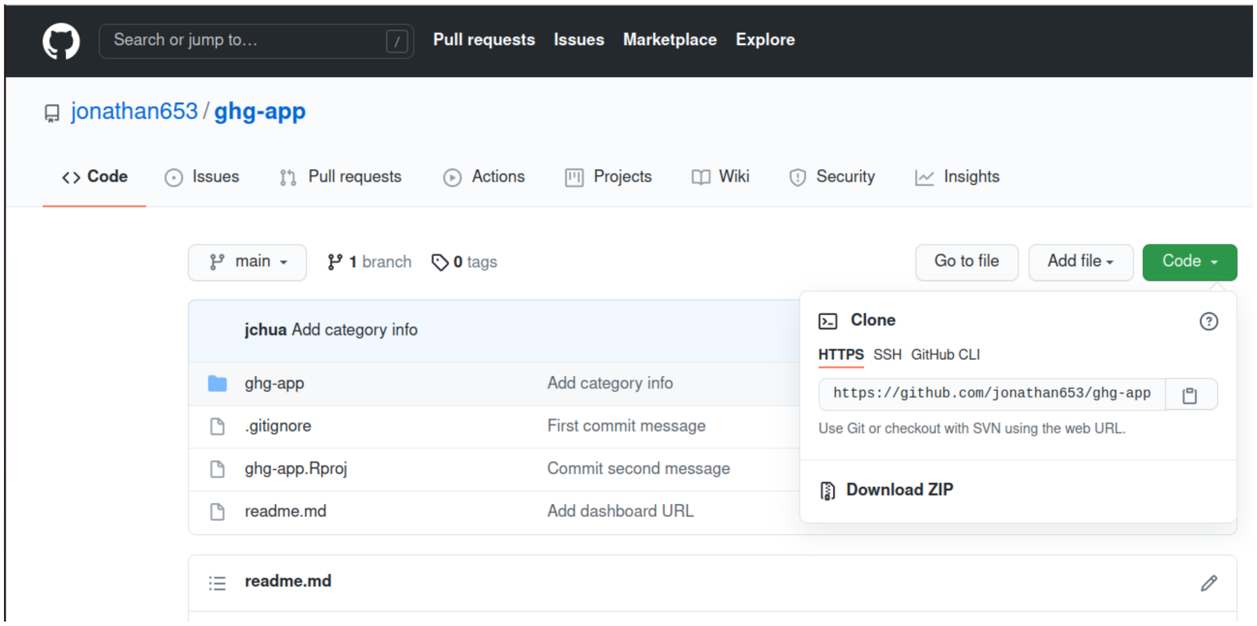
Craig Cliff, and his team work with other leaders and stakeholders across various departments at the university, in order to deliver sustainable progress in reaching Net Carbon Zero by 2030. They do this by reducing greenhouse gas emissions across all campuses and then offsetting the emissions that the university has not been able to eliminate. This app developed with Rshiny, will aid Craig in his delivery of workshops to educate university stakeholders on how universities can reduce greenhouse gas emissions.

This manual contains four sections. Software Set Up, Data, Application Workflow and User Testing + Future Expansion. In these sections you will find the information required to set up necessary software. Update datasets, equations, and graphing. These sections will also explain the interactions between reactive components, and additional features which could be developed. This manual assumes basic knowledge of R, HTML and git, and is intended to be read alongside the user manual written by Johnathan Chua. Jonathan Chua is the front end developer for this project. Brett Jones and Monique Arron are the back end developers. Brett has been focused on the Data, and Monique on Reactivity.

1. Software Set up

1.a Clone git Repository

Create/log in to your github account. Navigate to https://github.com/jonathan653/ghg-app.



jchua Add category info

ghg-app	Add category info
.gitignore	First commit message
ghg-app.Rproj	Commit second message
readme.md	Add dashboard URL

```
https://github.com/jonathan653/ghg-app
```

Use Git or checkout with SVN using the web URL.

readme.md

Figure 1a. Screenshot of ghg-app git repository cloning url.

Clone the repository (figure 1a). If you need instructions for using github you can find them at https://guides.github.com/

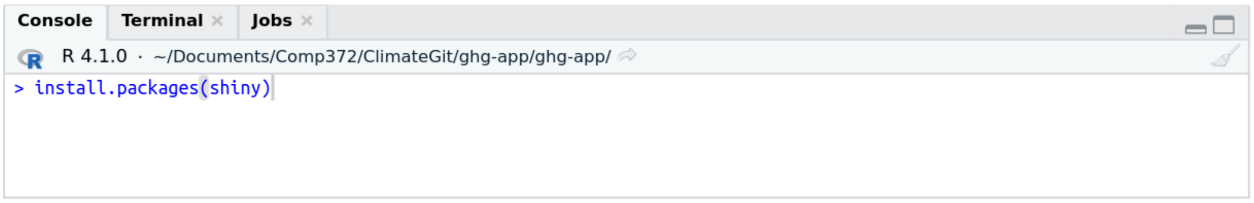
1.b R install

R is a free open source programming language licensed under the GNU General Public License (reference 2). The R language and environment is specialised for statistical computing and data visualisation. R can be downloaded from https://cran.stat.auckland.ac.nz/

1.c Rshiny package

"Shiny is an R package that makes it easy to build interactive web apps straight from R. You can host standalone apps on a webpage or embed them in R Markdown documents or build dashboards. You can also extend your Shiny apps with CSS themes, htmlwidgets, and JavaScript actions (RStudio, n.d.). "

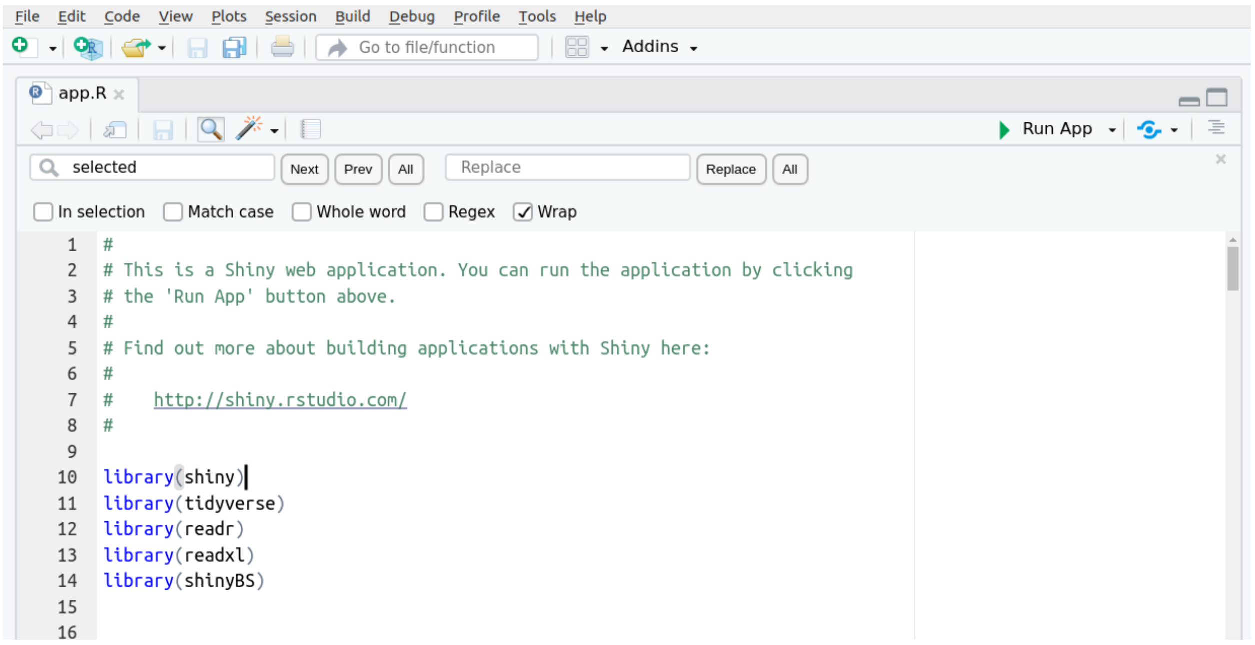
Rshiny is a package included in R but not automatically installed. Once you have R installed and opened R, set your working directory to your clone of the ghg-app repository. Install the shiny package using the console in R (figure 1b).



```
> install.packages(shiny)
```

Figure1b. Screenshot of install.packages(shiny) console command.

The Rshiny library has been declared at the top of the app.R file contained in the ghg-app git repository, along with the other libraries used by this shiny app (figure 1c).



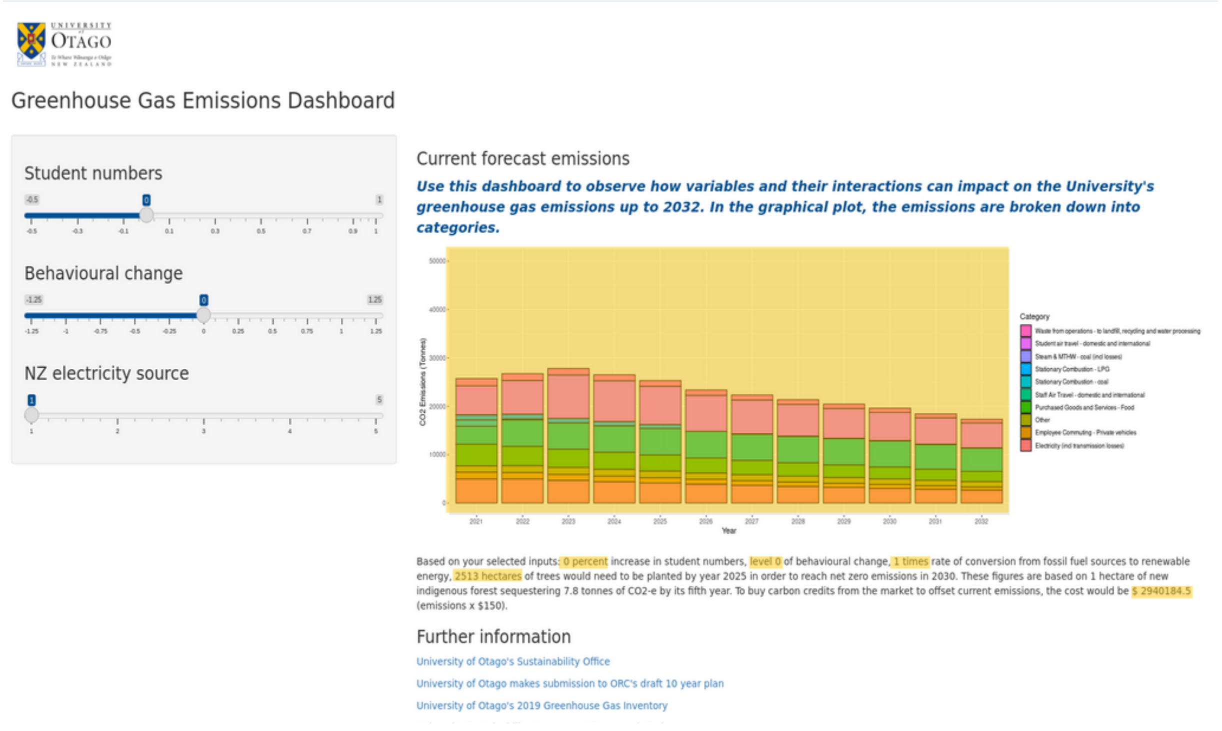
```
#
# This is a Shiny web application. You can run the application by clicking
# the 'Run App' button above.
#
# Find out more about building applications with Shiny here:
#
#    http://shiny.rstudio.com/
#

library(shiny)
library(tidyverse)
library(readr)
library(readxl)
library(shinyBS)

```

Figure 1c. Screenshot of libraries set-up in app.R

There are two major reactive functions from the shiny package used in this app. RenderPlot() and RenderText(). These functions respond to user input and update the graph and reactive text respectively (figure 1d). More details on Reactively can be found in section 3.


UNIVERSITY OF OTAGO
Greenhouse Gas Emissions Dashboard
Student numbers
Behavioural change
NZ electricity source
CO2 Emissions (Tonnes)
Year
Category
Waste from operations - to landfill, recycling and water processing
Student air travel - domestic and international
Steam & MTHW - coal (incl losses)
Stationary Combustion - LPG
Stationary Combustion - coal
Staff Air Travel - domestic and international
Purchased Goods and Services - Food
Other
Employee Commuting - Private vehicles
Electricity (incl transmission losses)

Current forecast emissions

Use this dashboard to observe how variables and their interactions can impact on the University's greenhouse gas emissions up to 2032. In the graphical plot, the emissions are broken down into categories.

Based on your selected inputs: 0 percent increase in student numbers, level 0 of behavioural change, 1 times rate of conversion from fossil fuel sources to renewable energy, 2513 hectares of trees would need to be planted by year 2025 in order to reach net zero emissions in 2030. These figures are based on 1 hectare of new indigenous forest sequestering 7.8 tonnes of CO2-e by its fifth year. To buy carbon credits from the market to offset current emissions, the cost would be $ 2940184.5 (emissions x $150).

Further information

University of Otago's Sustainability Office

University of Otago makes submission to ORC's draft 10 year plan

University of Otago's 2019 Greenhouse Gas Inventory

Figure 1d. Screenshot of app.R dashboard with reactive outputs from renderplot() and renderText() highlighted.

1.d Additional packages and Libraries

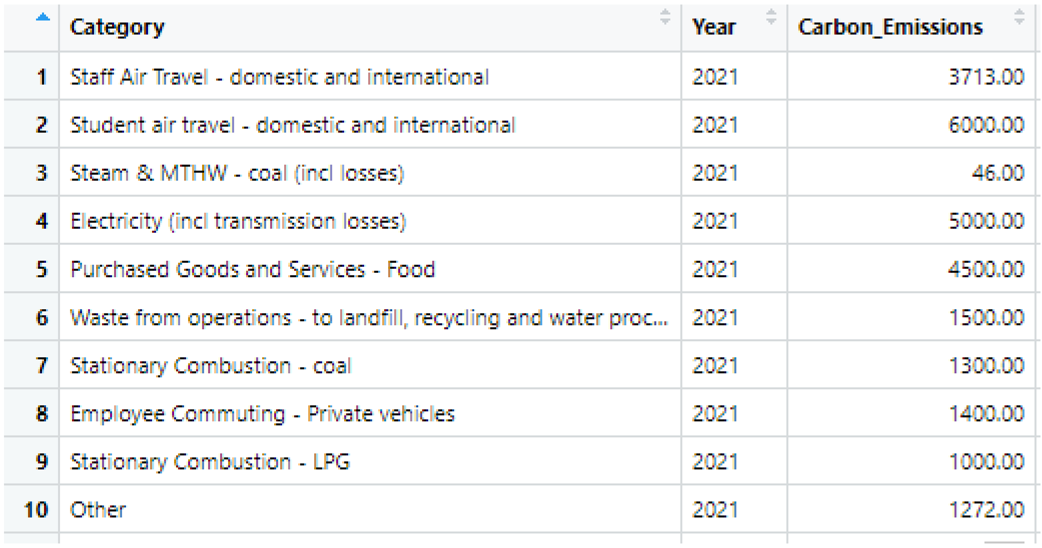
Install the following packages as seen in section 1.c. These packages provide additional functionality required to run the ghg-app.

- Tidyverse - A package which contains a set of packages designed to augment data science workflows. These packages share grammar and data structures.
- Dplyr - Part of the tidyverse package. Dplyr provides functions for data manipulation such as select(), filter(), mutate(), group_by() and summarise(). These functions have been used to create the tables from which the interactive graph is plotted.
- Readxl - Enables R to read and import Microsoft XL files, such as the xlsx file which contains our datasets.

- shinyBS - Adds bootstrap components to R. Provides additional reactive functionality to shiny apps.

2. Data

2.a The variables



	Category	Year	Carbon_Emissions
1	Staff Air Travel - domestic and international	2021	3713.00
2	Student air travel - domestic and international	2021	6000.00
3	Steam & MTHW - coal (incl losses)	2021	46.00
4	Electricity (incl transmission losses)	2021	5000.00
5	Purchased Goods and Services - Food	2021	4500.00
6	Waste from operations - to landfill, recycling and water proc...	2021	1500.00
7	Stationary Combustion - coal	2021	1300.00
8	Employee Commuting - Private vehicles	2021	1400.00
9	Stationary Combustion - LPG	2021	1000.00
10	Other	2021	1272.00

There are ten variables that makeup the total carbon emissions figures used in this project. These categories were categorized by Deloittes in their annual greenhouse gas emissions report for the University of Otago.

1. Staff Air Travel - domestic and international
   a. Emissions generated from staff air travel, both international and domestic. (11.3.1, Greenhouse Gas Inventory, 2019)

2. Student air travel - domestic and international
   a. Emissions generated from student air travel, both international and domestic. (personal communication, 21 July, 2021[1])

3. Steam & MTHW - coal (incl losses)
   a. Steam and medium temperature hot water (MTHW) generated from coal power. This includes losses generated in transmission and distribution. (10.2.1, Greenhouse Gas Inventory, 2019)

4. Electricity (incl transmission losses)
    a. The emissions generated from the production of electricity. This includes losses generated in transmission and distribution. (10.1.1, Greenhouse Gas Inventory, 2019)

5. Waste from operations - to landfill, recycling and water processing
    a. The emissions generated from waste to landfill, recycling and water processing. “Emissions include meat (excluding poultry), poultry, pulp and paper, wine, dairy processing.” (Section 9.3, table 65, MfE Guidance, 2020)

6. Purchased Goods and Services - Food
    a. “The emissions resulting from the purchase of food fall into two main categories: Food for consumption by students in residential colleges and food for sale in retail outlets or events.”(11.10.1, Greenhouse Gas Inventory , 2019)
7. Stationary Combustion - coal
    a. ‘Stationary combustion fuels are burnt in a fixed unit or asset, such as a boiler. Emissions occur from the combustion of fuels from sources owned or controlled by the reporting organisation.’ (9.2.1, Greenhouse Gas Inventory, 2019)

8. Employee Commuting - Private vehicles
    a. Staff commuting to and from the Otago University campus has been based on two primary sources of data - census mapping and the 2019 staff travel survey. (11.7.1, Greenhouse Gas Inventory, 2019)

9. Stationary Combustion - LPG
    a. ‘Stationary combustion fuels are burnt in a fixed unit or asset, such as a boiler. Emissions occur from the combustion of fuels from sources owned or controlled by the reporting organisation.’ (9.2.1, Greenhouse Gas Inventory, 2019)

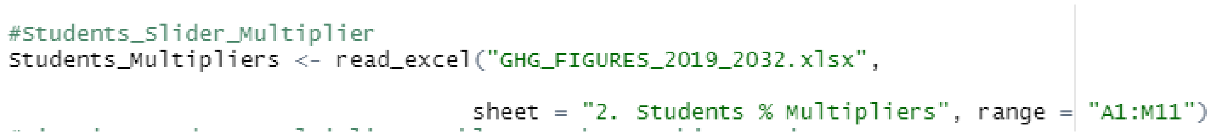
10. Other
    a. The other column is a combination of all factors contributing to the University of Otago’s greenhouse gas emissions that aren’t deemed to be significant enough to have an individual category of their own. This is subject to change in future versions of this application.

        The categories included in other are:

        i. Student commuting

ii. Business Travel - accommodation
iii. Steam & MTHW - biomass (incl losses)
iv. Business Travel - mileage, taxis and shuttles
v. Fugitive Emissions - refrigerants
vi. Mobile Combustion - diesel, petrol, pcard & marine
vii. Stationary Combustion - biomass
viii. Purchased Goods and Services - water
ix. Stationary Combustion - diesel
x. Employee Commuting -public transport
xi. Construction & demolition

2.b Importing Data



```
#Students_Slider_Multiplier
Students_Multipliers <- read_excel("GHG_FIGURES_2019_2032.xlsx",

                                   sheet = "2. Students % Multipliers", range = "A1:M11")
```

The data frames created and imported for this project all come from one table, 'GHG_FIGURES_2019_2032.xlsx'. There are six data frames used for this project. The first five data frames referenced are used to create the sixth data frame, titled 'GHG_Complete_Table'. The five tables imported all share the same cleaning methods, table structure and manipulations.

1. Base_Scenario
   a. Is a data frame created to import the table that stores the base scenario figures. The base scenario is the University of Otago's current greenhouse gas emissions forecast. (personal communication, 15 July, 2021[1])

2. Students_Multipliers
   a. Is a data frame created to import the table for the Students_Multiplier column figures. This column is used for computing a part of the equation to derive the increase in student numbers.

3. Lobc_Multipliers (Level of behavioural change)

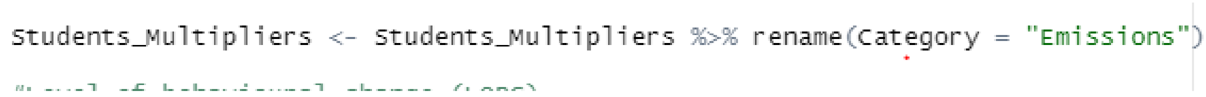
a. Is a data frame created to import the table for the Lobc_multiplier column figures used for deriving part of the equation for the changes in level of behavioural changes.

4. Electricity_Multipliers
    a. Is a data frame created to import the table for the Electricity_multiplier column figures used to derive part of the equation for the increase in the change to renewable resources (personal communication, 27 July, 2021[1]).

5. Scenarios
    a. Is a data frame created to import the table for the five scenarios columns that represent the rate of renewables change within New Zealand. (personal communication, 27 July, 2021[1])

6. GHG_Complete_Table
    a. Is a data frame created to compile five data frames into one, for the purpose of creating a renderplot, 'Base_Scenario_Graph').

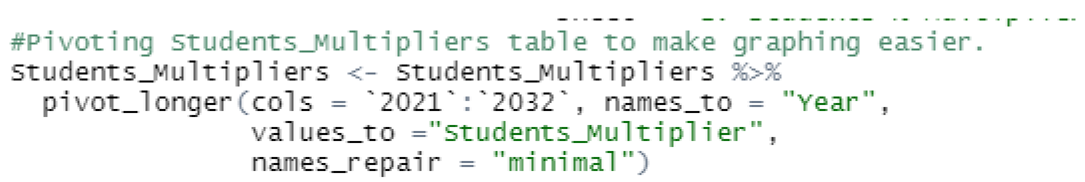
2.c Cleaning the Data



```
Students_Multipliers <- Students_Multipliers %>% rename(Category = "Emissions")
```

1. Category
    a. Renaming the 'emissions' category from the five imported xlsx files to 'category' to avoid confusion in the data manipulation and visualisation stages.

2.d Table structure



```
#Pivoting Students_Multipliers table to make graphing easier.
Students_Multipliers <- Students_Multipliers %>%
  pivot_longer(cols = `2021`:`2032`, names_to = "Year",
               values_to ="Students_Multiplier",
               names_repair = "minimal")
```

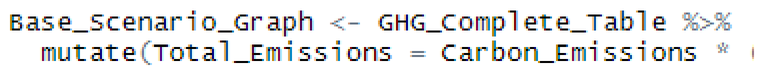
1. Pivot table
    a. Columns ‘Year’ and ‘Carbon_Emissions’ were created by pivoting the data frames for Base_Scenario, Students_Multipliers, Lobc_Multipliers and Electricity_Multipliers. Having the columns merged into one column with year entries is necessary to create a stacked column graph.
2. Year
    a. The year category was created from the numerical year columns (2021:2032) from the imported tables for data frames Base_Scenario, Students_Multipliers, Lobc_Multipliers and Electricity_Multipliers.
3. Category
    a. Category contains all ten of the variable names from section 3.a and is a feature of all six tables.
4. Carbon_Emissions
    a. Column created in the Base_Scenario table after pivot_wider applied to create the year column. Carbon_Emissions stores the values for the C02 emissions from category and year.
5. Multipliers

   The three multipliers respectively are the figures supplied by Otago University's sustainability office. These figures are multiplied against the base scenario figures to manipulate the column graph (see figures 3.f)

    a. There are three ‘multipliers’ columns

i. Students_Multipliers data frame. To be applied to the ‘student numbers’ part of the final equation.
ii. Electricity_Multipliers - Electricity_Multipliers data frame. To be applied to the ‘electricity’ part of the final equation.
iii. Lobc_Multipliers - Lobc_Multipliers data frame. To be applied to the ‘levels of behavioural change’ section of the final equation.

3.e Manipulating Data



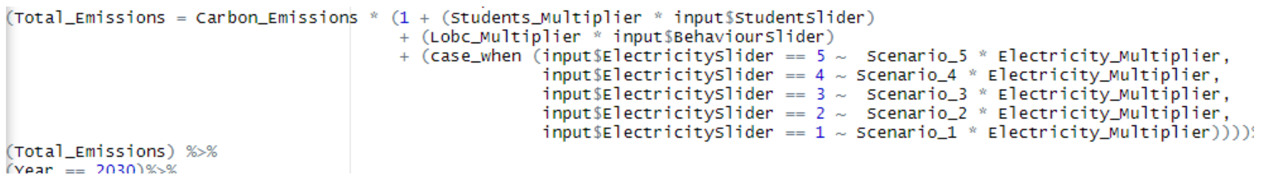
```
Base_Scenario_Graph <- GHG_Complete_Table %>%
  mutate(Total_Emissions = Carbon_Emissions *
```

The dplyr package was used to manipulate the data within the data frames.

1. Select, Filter, Mutate
   a. Select - Select columns you wish to use (removes unwanted columns).
   b. Filter - Filter data frame by specific feature. E.g if you wanted only results from a specific year, you can apply filter == (year).
   c. Mutate - Used to generate new columns within a data frame. E.g if you wanted to create a new column from an existing column, mutate(new column name = old column name)

2. Pipe to equations
   a. The %>% symbol is a pipeline operator. It is the dplyr package's way of telling R to change the data in a specified way. E.g A_Data_Frame %>% function. This notation is used frequently throughout the script. Note: Install the ‘Tidyverse’ package before running any chunk containing a %>%.

2.f Equations



```
(Total_Emissions = Carbon_Emissions * (1 + (Students_Multiplier * input$StudentSlider)
                                          + (Lobc_Multiplier * input$BehaviourSlider)
                                          + (case_when (input$ElectricitySlider == 5 ~  Scenario_5 * Electricity_Multiplier,
                                                        input$ElectricitySlider == 4 ~ Scenario_4 * Electricity_Multiplier,
                                                        input$ElectricitySlider == 3 ~  Scenario_3 * Electricity_Multiplier,
                                                        input$ElectricitySlider == 2 ~  Scenario_2 * Electricity_Multiplier,
                                                        input$ElectricitySlider == 1 ~ Scenario_1 * Electricity_Multiplier))))
(Total_Emissions) %>%
(Year == 2030)%>%
```

This section encompasses all the formulas used to create the equations used in this script.

Total_Emissions is the variable that is used on the y axis in 'Base_Scenario_Graph'. Depending on changes applied by a user moving the sliders, Total emissions will change how high or low the stacked bars go. The formula below is generated using data from the combined data frame 'GHG_Complete_Table'. The formula that makes up the equation for the Total_Emissions variable is:

Total_Emissions = Carbon_Emissions * (1 + (Student_Multiplier * input$StudentSlider) + (Lobc_Multiplier * input$BehaviourSlider) + (case_when (input$ElectricitySlider == 5 ~ Scenario_5 * Electricity_Multiplier, input$ElectricitySlider == 4 ~ Scenario_4 * Electricity_Multiplier, input$ElectricitySlider == 3 ~ Scenario_3 * Electricity_Multiplier, input$ElectricitySlider == 2 ~ Scenario_2 * Electricity_Multiplier, input$ElectricitySlider == 1 ~ Scenario_1 * Electricity_Multiplier))))

An explanation of what the individual parts of this formula do has been broken down into sections and is provided below:

1. Carbon_Emissions.
    a. Carbon emissions is a column containing the base scenario figures of the universities current GHG emissions pathway, and is the default state for the when opening the web tool. Mathematically, it works like this:
        i. If no changes are made to the sliders, and the graph is showing the base scenario, Total_Emissions = Carbon_Emissions * (1)
2. Student_Multipliers,
    a. Multiplier values for the student numbers. I.e, their weighted effect on the base scenario. Isolating their effects on Total_Emissions looks like this:
        i. Total_Emissions = Carbon_Emissions * (1 + Student_Multipliers), i.e the formula adds 1 to every value in the column. An example of what this might look like if we isolated one cell from Student_Multipliers

and its corresponding row value from the Carbon_Emissions column would be Total_Emissions = 5480 * (1.25). This is happening for every pair of values in the two columns and Total_Emissions is the sum of those multiplications.

3. input$StudentSlider.
    a. Is the change in the student slider variable on the formula. This change is made by the user adjusting the student slider. Expanding on the example from above. Assume we move the slider from position 0 to .1 (a 10% increase in student numbers):
        i. Total_Emissions = Carbon_Emissions * (1 + (Student_Multipliers(0.25) * StudentSlider(0.1)) Total_Emissions = 5480 * (1.025). Like above, Total_Emissions is the sum of all the rows in Carbon_Emissions, Student_Multipliers, and StudentSlider being manipulated like in the example.
4. Lobc_Multipliers.
    a. Is the multiplier used to determine the weights for a change in behaviours effect on the base scenario. This Multiplier works in the same way as the multiplier Student_Multiplier, so when visualising its interaction on the formula, simply substitute it into the same formula with its corresponding slider.
5. input$BehaviourSlider
    a. The formula works in the same way as described for the StudentSlider, except with the Lobc_Multiplier applied instead..
6. ElectricitySlider
    a. Operates slightly differently than the other two sliders on Total_Emissions. Because there are 5 different scenarios at which the electricity grind could transition to fully renewable sources in New Zealand, the scenarios need to be factored in. This is done in Rshiny using a case_when function, which operates the same way and if_else statement works in other languages. Below is an example of how a scenario change will impact the Electricity slider and Total_Emissions (isolating just the ElectricitySlider, Electricity_Multiplier and Scenario).

i. Total_Emissions = Carbon_Emissions *
(1 + (Electricity_Multipliers(-0.25) *
(ElectricitySlider(-0.25) *Scenario_5(0.25))

7. Electricity_Multipliers
    a. Works the same way as the other multipliers described above.
8. Scenarios
    a. Scenarios effect is described in the example above.

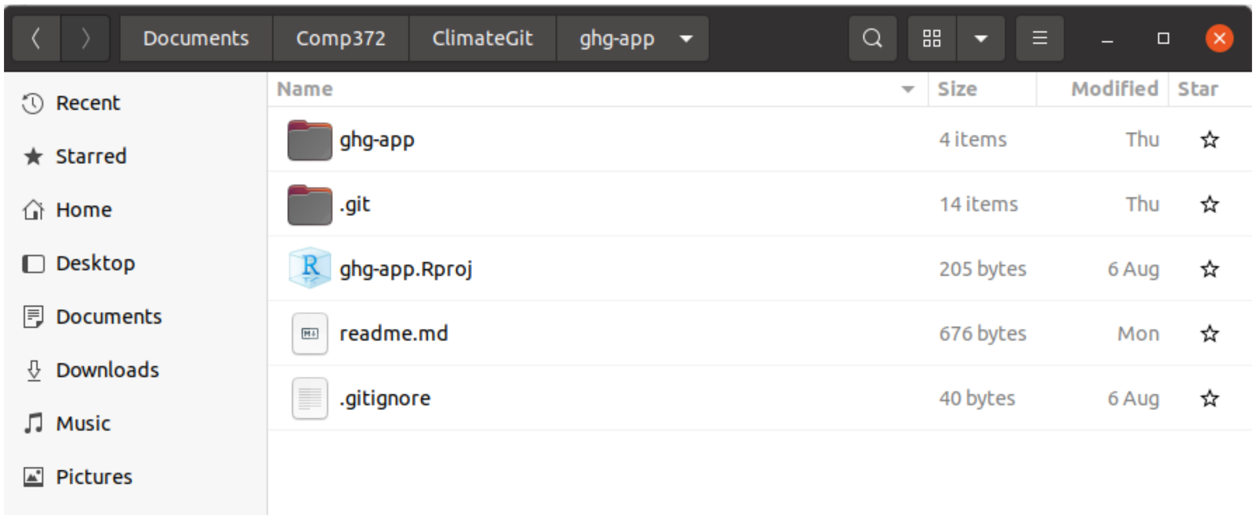
The following two formulas apply to the text section of the web tool.

9. Total emissions / 7.8 (hectares needed to be planted).
    a. In the text below the graph, there is a section that automatically updates to let the user know how many hectares of trees need to be planted in order to offset the figures displayed on the graph. It is simply Total_Emissions/7.8. equals the amount of hectares needed to offset one tonne of carbon emissions.
10. Total emissions * $150 (carbon credits).
    a. This is also in the text below the graph, it automatically updates to show the cost of purchasing carbon credits to offset the carbon, and is hard coded at $150 dollars per tonne. See future expansion for more details on this. The formula is simply Total_Emissions * 150

3. Application workflow

3.a File Structure

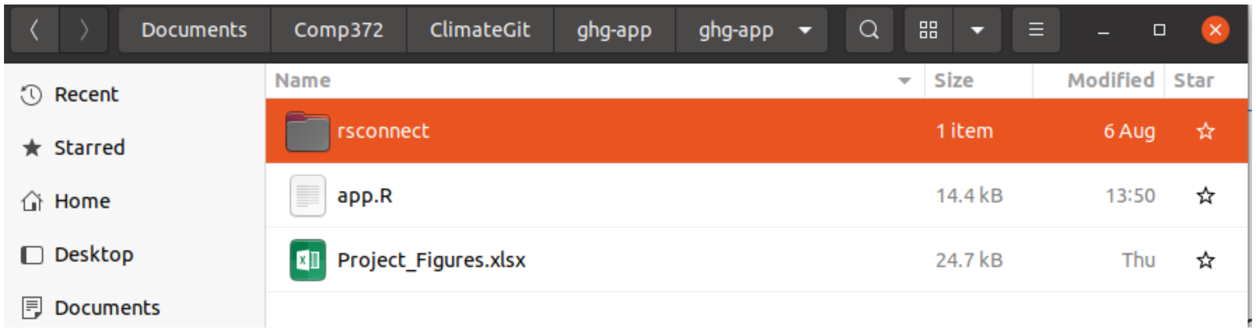
- Ghg-app folder - Contains all required files (figure 3a)



Name	Size	Modified	Star
ghg-app	4 items	Thu	☆
.git	14 items	Thu	☆
ghg-app.Rproj	205 bytes	6 Aug	☆
readme.md	676 bytes	Mon	☆
.gitignore	40 bytes	6 Aug	☆

Figure 3a. Screenshot of file structure inside the ghg-app folder.

- ghg-app folder 2 (figure 3b) contains:



Name	Size	Modified	Star
rsconnect	1 item	6 Aug	☆
app.R	14.4 kB	13:50	☆
Project_Figures.xlsx	24.7 kB	Thu	☆

Figure 3b. Screenshot of ghg-app folder contents

  - Rsconnect folder → shinyapps.io → otago-university → ghg-app.dcf

    Deploys app on shiny server.

  - app.R - application file. Has to be named app.R and be located inside the project folder in order to work.
  - ProjectFigures.xml - Where the datasets for import are kept.
- .git folder - this is where the git files are. If you need more information please read the official documentation.
- ghg-app.Rproj - Project file that can be used as a shortcut for opening the project directly. Is a dependency of app.R.

- Readme.md - Readme file
- .gitignore - Tells git which files should not be tracked

3.b Ui

UI is one of three components of shiny app architecture, contained within an app.R file. The UI is the front end containing the html code which will display the webpage that the user sees when they follow a link to a shiny app. The UI code is contained within **ui <- fluidPage()**

Within the UI are widgets. Widgets such as sliders users can interact with, take an input which is named and can be called in the server to update render functions.

3.c Server

The Server component is the backend containing the R code necessary for the shiny app. The server code is contained within **server <- function(input, output) {}**

And starts immediately after the ui <-fluidPage() close bracket.

It is within these brackets that everything the user doesn't see happens: Data import, manipulation, plotting, functions etc.

The code that should be outside the ui and server is **shinyApp(ui = ui, server = server)** after the server close bracket. This line of code is the third component of a shiny app. It runs the app.

3.d Reactivity

All reactive outputs require an input. In this case the inputs are values taken from three sliders. These inputs are used by reactive expressions to update the data required to generate the output, figure 3.c.

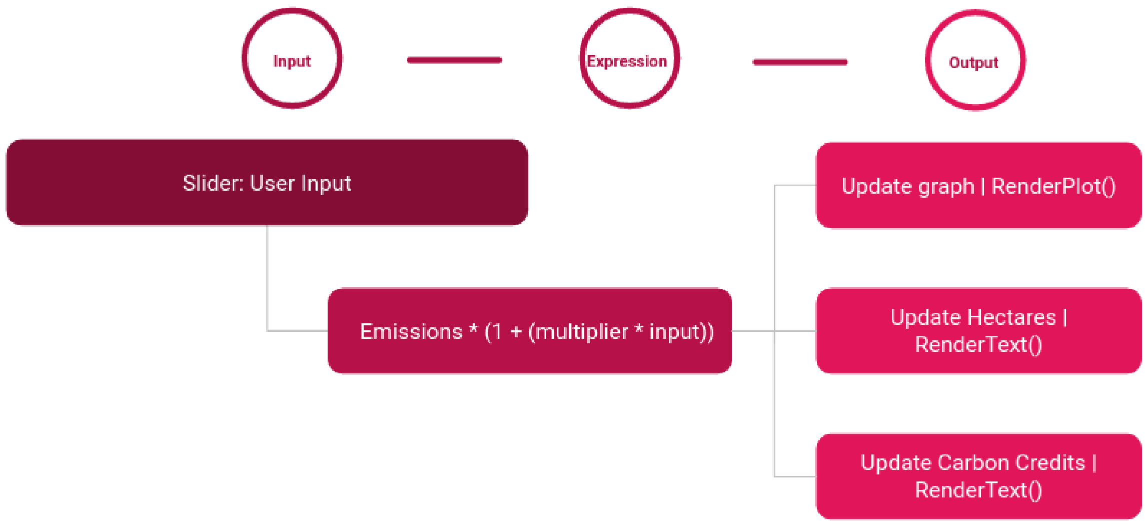
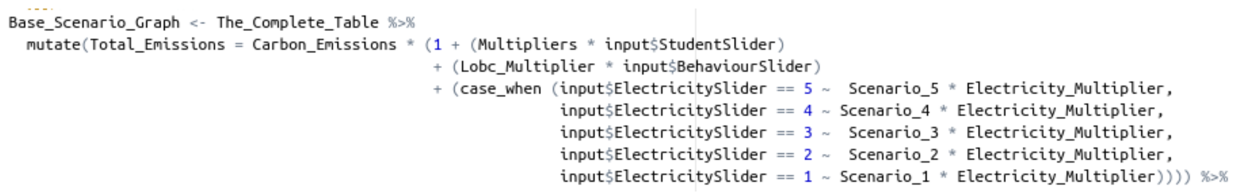

Input
Expression
Output
Slider: User Input
Emissions * (1 + (multiplier * input))
Update graph | RenderPlot()
Update Hectares | RenderText()
Update Carbon Credits | RenderText()

Figure 3.c Flowchart of ideal interactions within the app.

Ideally the middle step of the reactive expression would have been encapsulated as a reactive function which all render functions could call. Unfortunately due to the limited expertise and time available to the team we did not manage to implement this part of the workflow optimally. Please see section 4.h.1.a for discussion on how to correct this.

We did however succeed in updating the render outputs by piping code within them. Figure 3.d shows the code which would ideally be encapsulated within its own function.



```
Base_Scenario_Graph <- The_Complete_Table %>%
  mutate(Total_Emissions = Carbon_Emissions * (1 + (Multipliers * input$StudentSlider)
                                              + (Lobc_Multiplier * input$BehaviourSlider)
                                              + (case_when (input$ElectricitySlider == 5 ~  Scenario_5 * Electricity_Multiplier,
                                                             input$ElectricitySlider == 4 ~ Scenario_4 * Electricity_Multiplier,
                                                             input$ElectricitySlider == 3 ~  Scenario_3 * Electricity_Multiplier,
                                                             input$ElectricitySlider == 2 ~  Scenario_2 * Electricity_Multiplier,
                                                             input$ElectricitySlider == 1 ~ Scenario_1 * Electricity_Multiplier)))) %>%
```

Figure 3.d screenshot of repeatedly piped code

3.e Sliders

Sliders are shiny widgets which have a minimum, maximum and starting value. Sliders allow the user to change the input. When they do this any functions in the server which call the slider input will update their output to display updated information.

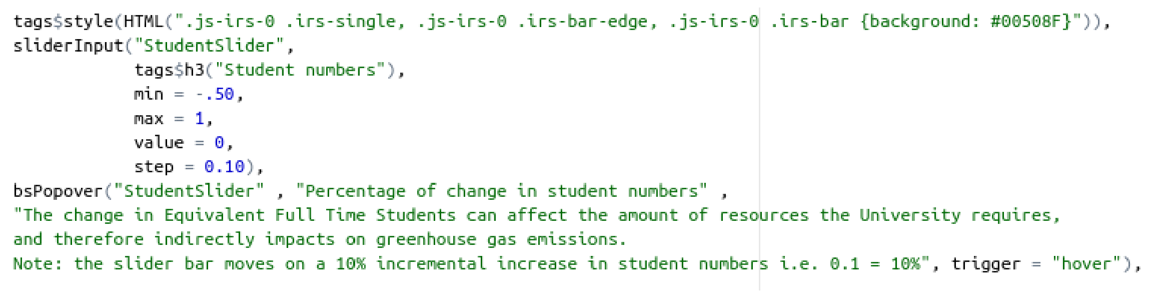
Each slider has a line of html code above it (figure 3.e) setting out it's aesthetics. Below the shiny code for each slider is a bsPopover() function which takes the slider name, a title, pop over text and a trigger condition, as parameters. This method displays a pop over text box when the user hovers their mouse over the slider. Here we have the pop over box text explaining what the Student numbers slider pertains to.

In figure 3.e **sliderInput("StudentSlider")** contains the input variable **StudentSlider** which is updated by the user's changes and called by functions in the server.

To change the minimum and maximum values of a slider, simply change the values for **min =** ______, and **max =** . Note that this slider deals with percentages. 1 = 100%, 0.10 = 10% and -.50 = -50%.

Value = represents the default value that the app will open on. Currently set to zero, or no change.

Step = the points to which a user can move a slider. Make sure that any change you make to this number is divisible by both your min and max values, or your user may not be able to return the slider to its starting position of zero.



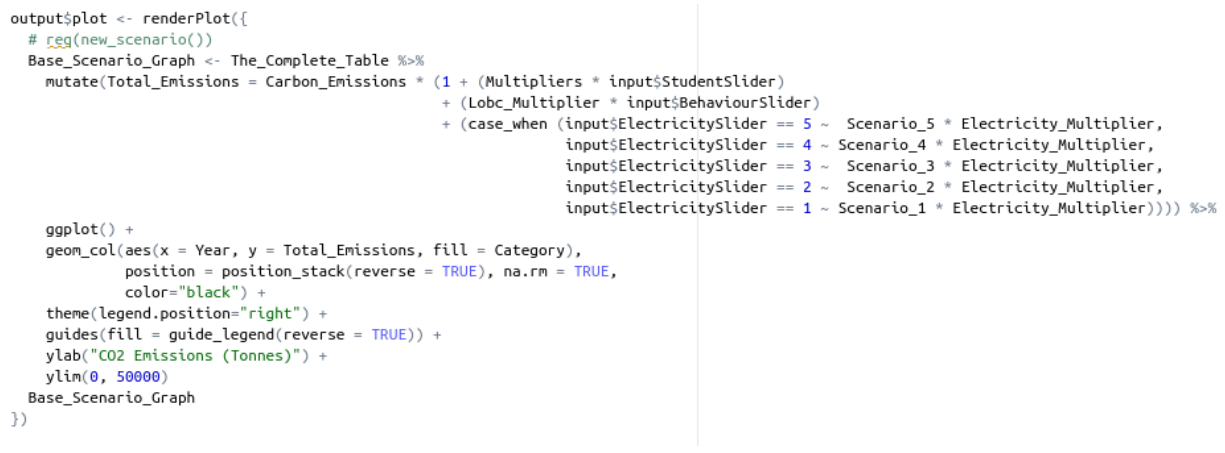
```
tags$style(HTML(".js-irs-0 .irs-single, .js-irs-0 .irs-bar-edge, .js-irs-0 .irs-bar {background: #00508F}")),
sliderInput("StudentSlider",
            tags$h3("Student numbers"),
            min = -.50,
            max = 1,
            value = 0,
            step = 0.10),
bsPopover("StudentSlider" , "Percentage of change in student numbers" ,
"The change in Equivalent Full Time Students can affect the amount of resources the University requires,
and therefore indirectly impacts on greenhouse gas emissions.
Note: the slider bar moves on a 10% incremental increase in student numbers i.e. 0.1 = 10%", trigger = "hover"),
```

Figure 3.e Screenshot of UI student numbers code

3.f Plotting

output$plot <- renderPlot({}) contains the code required to generate the graph. Within **mutate()** the slider input values are multiplied by multipliers stored in tables, figure 3.f. In the case of the electricity slider **case_when()** is used to determine which column of the table to multiply by. For more details on data structures refer to section 2.

The results of **mutate()** are used to plot a stacked bar chart using **ggplot()** as normal in R, results shown in figure 3.g.



```r
output$plot <- renderPlot({
  # req(new_scenario())
  Base_Scenario_Graph <- The_Complete_Table %>%
    mutate(Total_Emissions = Carbon_Emissions * (1 + (Multipliers * input$StudentSlider)
                                                 + (Lobc_Multiplier * input$BehaviourSlider)
                                                 + (case_when (input$ElectricitySlider == 5 ~  Scenario_5 * Electricity_Multiplier,
                                                               input$ElectricitySlider == 4 ~ Scenario_4 * Electricity_Multiplier,
                                                               input$ElectricitySlider == 3 ~  Scenario_3 * Electricity_Multiplier,
                                                               input$ElectricitySlider == 2 ~  Scenario_2 * Electricity_Multiplier,
                                                               input$ElectricitySlider == 1 ~ Scenario_1 * Electricity_Multiplier)))) %>%
    ggplot() +
    geom_col(aes(x = Year, y = Total_Emissions, fill = Category),
             position = position_stack(reverse = TRUE), na.rm = TRUE,
             color="black") +
    theme(legend.position="right") +
    guides(fill = guide_legend(reverse = TRUE)) +
    ylab("CO2 Emissions (Tonnes)") +
    ylim(0, 50000)
  Base_Scenario_Graph
})
```

Figure 3.f Screenshot of renderPlot({}) code

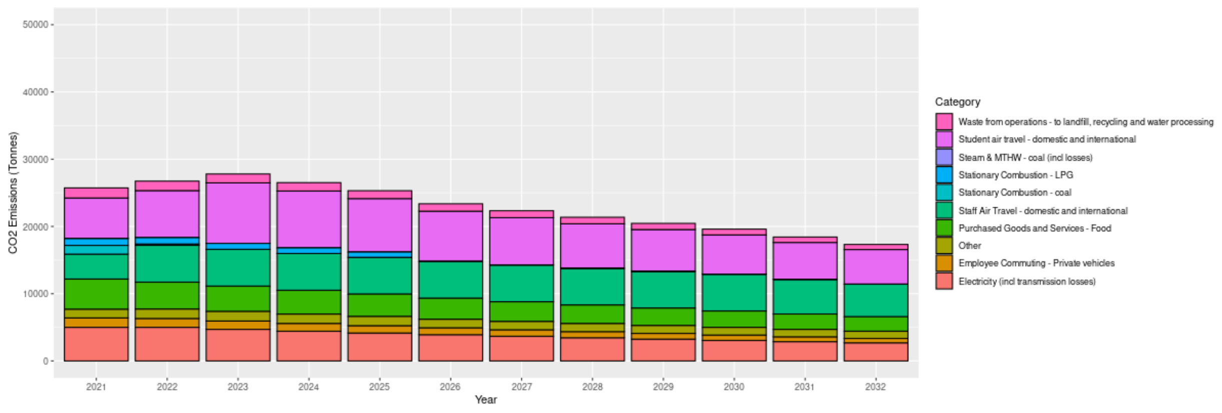
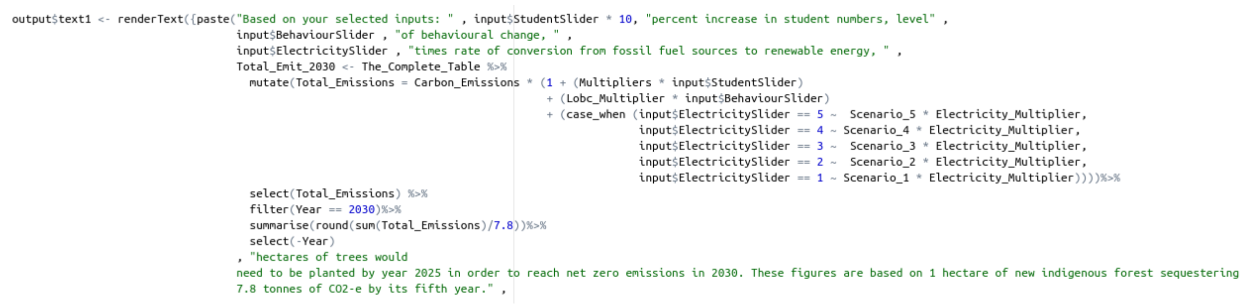


Figure 3.g Screenshot of the graph using default values.

3.i Text output

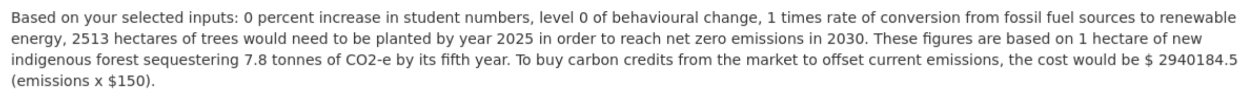
output$text1 <- renderText({})** prints text to the screen. In figure 3.h, the text in green is static, with the code in between updating based on user inputs. **input$StudentSlider * 10 reports the percentage of change in student numbers that the user has entered. i**nput$BehaviourSlider** and **input$ElectricitySlider** return the level of change entered on those sliders. The clunky piped code generating the **Total_Emit_2030** table could be replaced by a function whose results are plugged into **summarise(round(sum(Total_Emissions)/7.8))** to produce the number of hectares for planting by 2025 to achieve the University of Otago's net zero carbon emissions target.

See figure 3.i for the text seen by the user.



```
output$text1 <- renderText({paste("Based on your selected inputs: " , input$StudentSlider * 10, "percent increase in student numbers, level" ,
                              input$BehaviourSlider , "of behavioural change, " ,
                              input$ElectricitySlider , "times rate of conversion from fossil fuel sources to renewable energy, " ,
                              Total_Emit_2030 <- The_Complete_Table %>%
                                mutate(Total_Emissions = Carbon_Emissions * (1 + (Multipliers * input$StudentSlider)
                                                                      + (Lobc_Multiplier * input$BehaviourSlider)
                                                                      + (case_when (input$ElectricitySlider == 5 ~  Scenario_5 * Electricity_Multiplier,
                                                                                   input$ElectricitySlider == 4 ~ Scenario_4 * Electricity_Multiplier,
                                                                                   input$ElectricitySlider == 3 ~  Scenario_3 * Electricity_Multiplier,
                                                                                   input$ElectricitySlider == 2 ~  Scenario_2 * Electricity_Multiplier,
                                                                                   input$ElectricitySlider == 1 ~ Scenario_1 * Electricity_Multiplier))))%>%
                                select(Total_Emissions) %>%
                                filter(Year == 2030)%>%
                                summarise(round(sum(Total_Emissions)/7.8))%>%
                                select(-Year)
                              , "hectares of trees would
                              need to be planted by year 2025 in order to reach net zero emissions in 2030. These figures are based on 1 hectare of new indigenous forest sequestering
                              7.8 tonnes of CO2-e by its fifth year." ,
```

Figure 3.h Screenshot of a renderText() codeblock.



```
Based on your selected inputs: 0 percent increase in student numbers, level 0 of behavioural change, 1 times rate of conversion from fossil fuel sources to renewable
energy, 2513 hectares of trees would need to be planted by year 2025 in order to reach net zero emissions in 2030. These figures are based on 1 hectare of new
indigenous forest sequestering 7.8 tonnes of CO2-e by its fifth year. To buy carbon credits from the market to offset current emissions, the cost would be $ 2940184.5
(emissions x $150).
```

Figure 3.i Screenshot of renderText() output.

4. User Testing & Future Expansion

4.a User Feedback

1. Explain emissions categories better.
    a. Universal feedback from user testing. Things like MTHW have no meaning to a layman user.
2. Show years 2019 and 2020 on the graph.
    a. A few users asked us to include stacked emissions column data for the years 2019 and 2020, with the same reasoning - context. After a team discussion, while we agreed with the users sentiment, we didn't implement this change because it might be also perceived as using outdated data sources by other users.

4.b Changes Implemented

1. Adding a description of all the categories that make up the column graph.

2. Changing multiplier values. The client identified changes he would like to be displayed on the graph in relation to the effects of the multipliers on certain categories. These categories are:
    i. Electricity (incl transmission losses) (for NZ electricity source)
    ii. Waste to landfill from operations (for behaviour slider)

4.c Aspects to revisit

1. Adding 2019 and 2020 data to the graph.
    a. This feedback change could be implemented in the future.

4.d Additional sliders and variables

1. Levels of emissions reductions targets
    a. A slider with the predefined levels of emissions reductions targets as defined by the universities sustainability office.
2. Growth rates of insetting projects.
    a. Insetting projects do not have immediate impacts on reducing GHG emissions, but instead take time to grow to full size. Implementing a growth rates slider would help management and executives visualise insetting projects in relation to time effects on total GHG emissions reduction targets, to make more informed decisions.
3. Percentage of energy generated on site.
    a. A slider indicating the effects of a switch to generating energy on site.
4. Low emissions domestic air travel.
    a. A slider to change between scenarios of low emissions domestic air travel from 0 (our current position) to 3 (fully electric).
5. Local council investment in active transport.
    a. Increments on the slider could be levels similar to the levels of behavioural change slider.
6. Free public transport for staff and students
    a. A simple on/off slider would suffice to show effects on GHG emissions reductions targets.
7. Food Policy

    a. Increments on the slider could be levels like the behavioural change slider. Low change could be implementing meat free dinners throughout the week for the entire student and staff population, while high change could be a fully vegan diet for the entire student and staff population.

8. Student air travel offset
    a. Including a slider to show the effects of a student air travel offset could be a useful device to implement to show the significant impact it could have on the 2030 target. A simple on/off slider would suffice.
9. Staff (FTE) (% growth or decline)
    a. Full time equivalent staff numbers grow proportionally in relation to student numbers, but could potentially have its own slider with a similar scale to student numbers, ranging from 0 to 100% increase in 10% increments.
10. Staff air travel offset
    a. This would be similar to the students air travel slider; a simple off/on toggle slider to demonstrate the significant impact to meeting the carbon neutral 2030 target.

4.e Ability to use as framework by other universities/organisations

It is the authors aspiration that this tool and the documentation provided could serve as a platform for other universities to build a similar web tool to increase awareness of the scope of net neutral carbon emissions plans with the intention of influencing policy change at an executive level.

4.f CSS theme

With the R shiny package, it is possible to add a CSS theme. Therefore in future versions of this project, the developer could potentially add a theme that the University of Otago uses for their other applications.

4.h Optimisation

To optimise this tool, here are a few recommendations:

1. Enclose repeated code in reactive functions

   a. Currently in version 1 of this build, there is a lot of repeated code used, specifically in the renderplot. Building separate functions and calling them within the renderplot will increase the speed of the web tool.
2. Make it mobile friendly
   a. We identified through user testing that mobile devices do not display the web tool accurately. Making it mobile friendly could increase user engagement with the web tool.
3. Database
   a. In the design phase we had a database, but decided not to include it in the final design, simply because the data sets are so small it renders the database useless. If the datasets increase in size in the future, implementing a SQLite database might be worthwhile.
4. Add function to automatically update the price of carbon
   a. A function that webscrapes a carbon price tracker and displays the correct price would be a useful feature.
5. Make individual category for student commuting
   a. Student commuting emissions are up to five times larger than other categories within the other category. This could mean that it is worthwhile making Student commuting its own category.

Conclusions

This report is intended to serve as an instructional guide to help other universities create a GHG emissions web tool of their own. It gives an overview of the software setup used to create the web application, the data used and formulas applied, the applications workflow, the categories and sliders included in the current version, and the future expansions envisioned by the authors.

Both authors of this report are two of the three developers who worked on this project. Subsequently, they have a detailed knowledge of the development process of this web app. This project was a learning opportunity for the authors with a time constraint of five weeks to complete the project, as such, there is room for improvement within the back end. It is the authors hope that developers working on this web application in the future will refine and expand on the work of the original development team.

References

University of Otago. (n.d.). *Otago Clocktower* [Photo]. https://www.studyoptions.com/university-profiles/university-otago

Chua, J., Arron, M., & Jones, B. (2021, August 24). *GitHub - jonathan653/ghg-app*. GitHub. https://github.com/jonathan653/ghg-app

Github. (n.d.). *GitHub Guides*. Github.Com. Retrieved August 23, 2021, from https://guides.github.com/

University of Auckland. (n.d.). *The Comprehensive R Archive Network*. Cran.Stat.Auckland.Ac.Nz. Retrieved August 23, 2021, from https://cran.stat.auckland.ac.nz/

1991 Free Software Foundation Inc. (1991, June 2). *R: The R Project for Statistical Computing*. R-Project.Org. https://www.r-project.org/COPYING

RStudio. (n.d.). *Shiny*. Rstudio.Com. Retrieved August 24, 2021, from https://shiny.rstudio.com/

New Zealand Government. (2020). *Measuring Emissions: A Guide for Organisations*. enviroment.govt.nz. https://environment.govt.nz/assets/Publications/Files/Measuring-Emissions-Detailed-Guide-2020.pdf

Deloitte. (2020, October). *Greenhouse Gas Inventory - 2019*. Www.Otago.Ac.Nz. https://www.otago.ac.nz/sustainability/otago824241.pdf

Appendix

1. According to the 7th APA guide, no entry in the reference list is needed for personal communications, as they are unable to be retrieved. Therefore we the authors have not referenced sections of this report pertaining to the sections referenced in the report as '(personal communication, <date>) '