

Agricultural activity and water quality

Group:

Status 418: I'm a teapot

Henri Liswoyo

Alzen Punio

HanByeol Yang

Moses Velano

Agricultural Activity

General workflow:

- Collecting data
- Removing irrelevant columns
- Filtering out irrelevant rows
- Standardising values and column names
- Joining the table
- Plotting

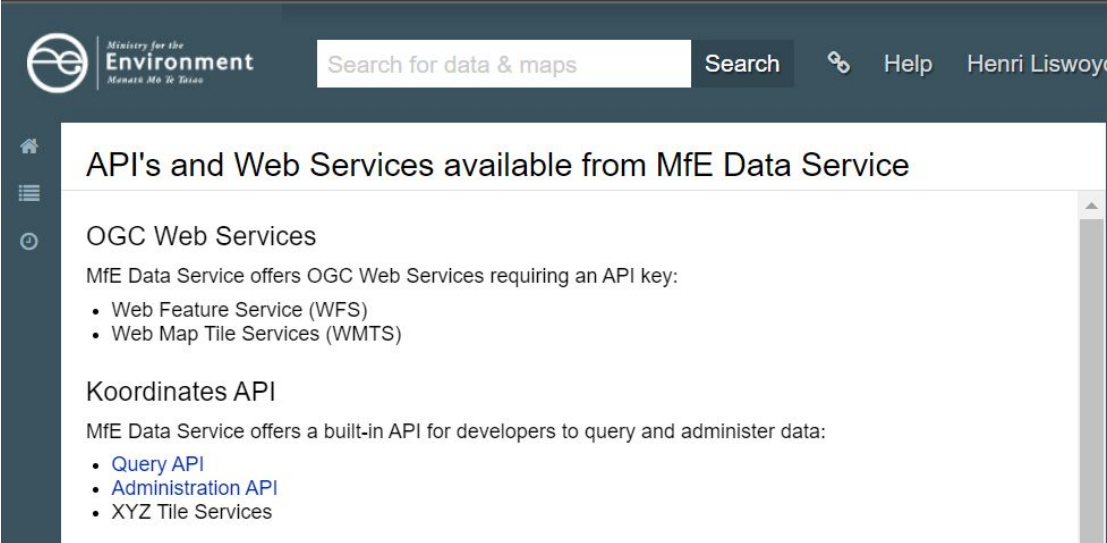
Agricultural Activity - Collecting Data

Sources:

Ministry for the Environment

StatsNZ

- Both are government sources
- Freely accessible to public
- API support



The screenshot shows the Ministry for the Environment (MfE) Data Service website. The header includes the MfE logo, the text "Ministry for the Environment" and "Minatū Mo Te Taiao", a search bar with the placeholder "Search for data & maps", and a "Search" button. The main content area is titled "API's and Web Services available from MfE Data Service". It lists "OGC Web Services" which require an API key, including "Web Feature Service (WFS)" and "Web Map Tile Services (WMTS)". It also lists "Koordinates API" which is a built-in API for developers to query and administer data, including "Query API", "Administration API", and "XYZ Tile Services".

Ministry for the Environment
Minatū Mo Te Taiao

Search for data & maps Search

API's and Web Services available from MfE Data Service

OGC Web Services

MfE Data Service offers OGC Web Services requiring an API key:

- Web Feature Service (WFS)
- Web Map Tile Services (WMTS)

Koordinates API

MfE Data Service offers a built-in API for developers to query and administer data:

- [Query API](#)
- [Administration API](#)
- XYZ Tile Services

Agricultural Activity - Collecting Data

API use:

```
# A function to get data from the Ministry for the Environment database API

get_data_from_mfe <- function(api_key, data_id){

  # creates a query url using inputs of api key and data-id number available on MfE website
  query <- glue('https://data.mfe.govt.nz/services;key={api_key}/wfs?service=WFS&version=2.0.0&request=GetFeature&typeName=')

  api_response <- GET(query) #gets the API response from the query

  data_xml <- read_xml(api_response) #reads the xml data from the api response

  data_parsed <- xmlParse(data_xml) #parses the data into an xml format that is readable in R

  data_df <- glue('///data.mfe.govt.nz:{data_id}') %>% # creating a node name to look for
  getNodeSet(data_parsed, .) %>% # looking at nodes with the name
  xmlToDataFrame(nodes = .) #turns the data within the given node into a data frame

  return(data_df) #returns the data frame
}
```

Agricultural Activity - Removing Irrelevant Columns

Livestock	Area	Year	Value	Flags
Total beef	Total New	As at June	5047848	
Total beef	Total New	As at June	4491281	
Total beef	Total New	As at June	4626617	

Agricultural Activity - Filtering Out Irrelevant Rows

Removing rows of aggregated data →

```
# Removing irrelevant datas by using the filter() function
```

```
latest_horti_pop <- latest_horti_pop %>%  
  filter(Area != "Total New Zealand") %>% # filtering unwanted areas  
  filter(Area != "Total North Island") %>% # filtering unwanted areas  
  filter(Area != "Total South Island") %>% # filtering unwanted areas
```

Calves born alive to beef heifers/cows	Northland	1994	136047
Calves born alive to beef heifers/cows	Northland	2002	107864
Calves born alive to beef heifers/cows	Northland	2003	104125
Calves born alive to beef heifers/cows	Northland	2004	94213

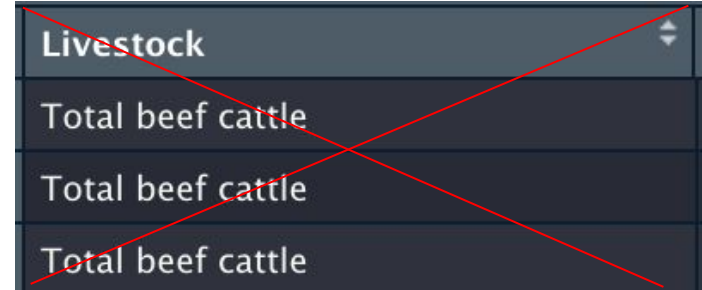
← Selecting relevant rows,
then removing all other rows

```
# Removing unnecessary data  
latest_livestock_pop_remove <- latest_livestock_pop %>% # creates new variable that removes irrelevant areas  
  filter(Livestock != "Total dairy cattle") %>% # filtering unwanted data  
  filter(Livestock != "Total beef cattle") %>% # filtering unwanted data  
  filter(Livestock != "Total sheep") %>% # filtering unwanted data  
  filter(Livestock != "Total deer") %>% # filtering unwanted data
```

```
# Extracting the necessary datas/variables we want  
latest_livestock_pop <- latest_livestock_pop[!(latest_livestock_pop$Livestock %in% latest_livestock_pop_remove$Livestock),]  
  
latest_livestock_pop
```

Agriculture Activities - Standardising values and column names

Renaming the column and the variables →
contained in it.

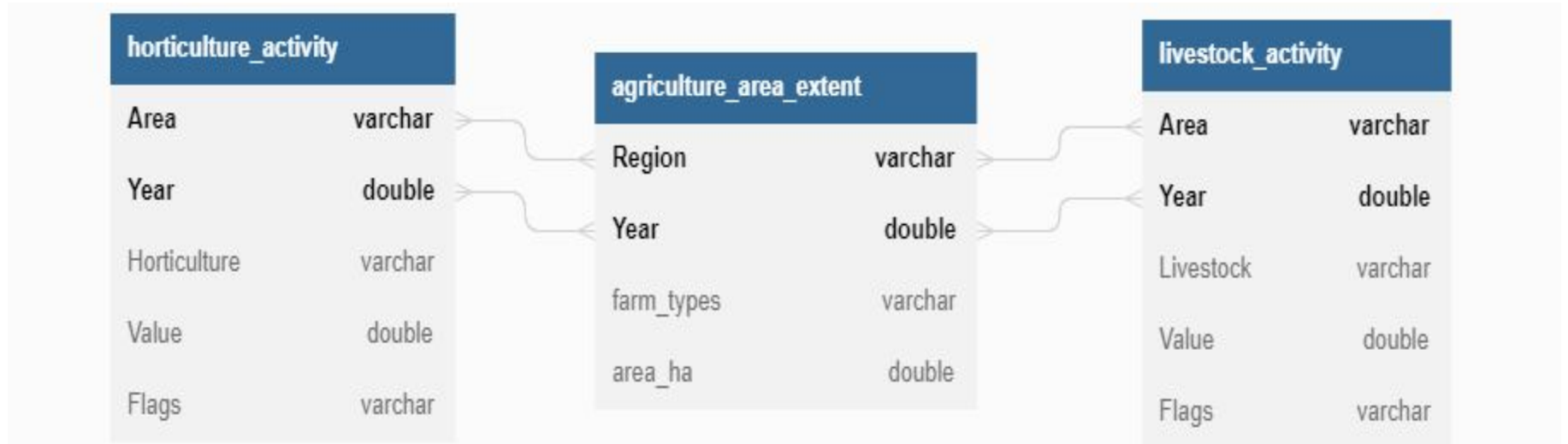


Livestock
Total beef cattle
Total beef cattle
Total beef cattle

```
# Renaming Variables.  
latest_livestock_pop <- latest_livestock_pop %>%  
  rename(Activity = "Livestock") # Renames Livestock to Activity  
  
latest_livestock_pop$Activity <- gsub("Total dairy cattle", "Dairy Cattle", as.character(latest_livestock_pop$Activity))  
latest_livestock_pop$Activity <- gsub("Total beef cattle", "Beef Cattle", as.character(latest_livestock_pop$Activity))  
latest_livestock_pop$Activity <- gsub("Total sheep", "Sheep", as.character(latest_livestock_pop$Activity))  
latest_livestock_pop$Activity <- gsub("Total deer", "Deer", as.character(latest_livestock_pop$Activity))
```

Agricultural Activity - Joining The Tables

From this...

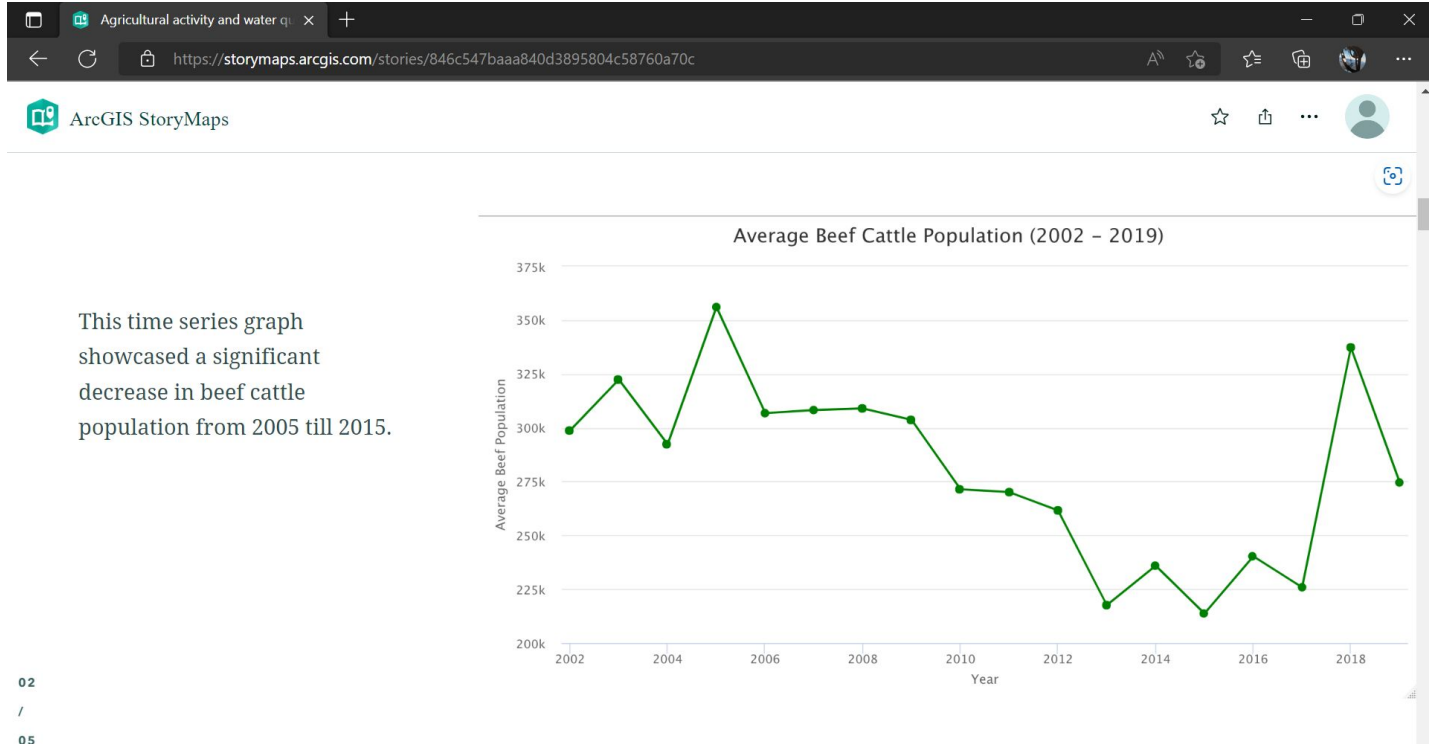


Agricultural Activity - Joining The Tables

... to this

```
Rows: 351
Columns: 26
$ Area                <chr>
$ Year                <dbl>
$ key                 <chr>
$ `Beef Cattle`       <dbl>
$ `Dairy Cattle`      <dbl>
$ Deer                <dbl>
$ Sheep               <dbl>
$ `Total apples (hectares)` <dbl>
$ `Total avocados (hectares)` <dbl>
$ `Total kiwifruit (hectares)` <dbl>
$ `Total olives (hectares)` <dbl>
$ `Total onions (hectares)` <dbl>
$ `Total potatoes (hectares)` <dbl>
$ `Total squash (hectares)` <dbl>
$ `Total wine grapes (hectares)` <dbl>
$ Total_Hectares      <dbl>
$ `Dairy area (ha)`   <dbl>
$ `Floriculture area (ha)` <dbl>
$ `Forestry area (ha)` <dbl>
$ `Fruit and berry area (ha)` <dbl>
$ `Grain growing area (ha)` <dbl>
$ `Nursery and turf area (ha)` <dbl>
$ `Other area (ha)`   <dbl>
$ `Other Livestock area (ha)` <dbl>
$ `Sheep and Beef area (ha)` <dbl>
$ `Vegetable growing area (ha)` <dbl>
```

Agriculture Activities - Plotting



Water Quality

General workflow

- Data collection
- Data wrangling
 - Conversion of river sites' latitudes and longitudes
 - Making sense out of the NAs
 - Standardising the units and region name
 - Selecting and renaming relevant columns
- Plotting
 - Overall plots
 - Proportion plots
 - Maps (sites exceeded the threshold and average nitrogen concentration across New Zealand)
- Data model

Data Model - Water Quality

river_quality	
Region	varchar
S_ID	varchar
Year	double
MeanVal	double
Indicator	varchar

river_srcs	
Region	varchar
S_ID	varchar
Latitude	double
Longitude	double

groundwater_quality	
Region	varchar
WellName	varchar
Year	double
MeanVal	double
Indicator	varchar

groundwater_sites	
Region	varchar
WellName	varchar
Latitude	double
Longitude	double

Challenges of our project

- GitHub
- Web scraping and API
- Plotting challenges
- Extreme values for water quality
- Dashboarding tool

The end product

Graphs and maps publicly accessible through a story map webpage

<https://storymaps.arcgis.com/stories/846c547baaa840d3895804c58760a70c>