

Wk10 Assessment

TASK

Question 1.

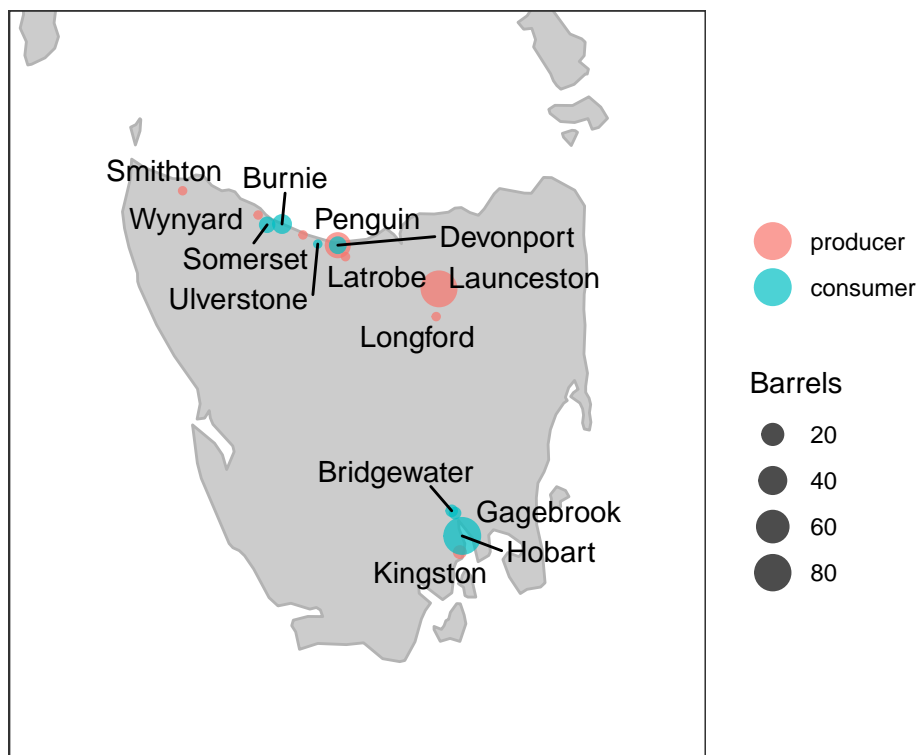
Create two maps using the data contained in `rnaturalearth`. One must have a map extent that is smaller than the entire world (e.g. only shows one continent, or is bounded in some other way). Both must have polygons that are coloured according to some aspect of the data. In the workshop you created the entire world coloured by the square root of the total population, so you cannot use population data for this question.

For each of your plots, represent that same data using at least one other plot type that we learned earlier in the unit, identify a target audience and an intended message for your visualisation.

Question 2.

Reproduce the following map using the locations and transport datasets provided to you in the workshop files on Blackboard.

Whiskey sales in Tasmania
in the 1890s



This is a difficult question, for which you'll need to draw from your past experiences in this unit wrangling data and changing elements of the `ggplot` outputs. If you cannot get all the way to the final figure, for partial marks you can submit the closest figure you can get to and briefly explain: (1) the roadblock you could not pass and (2) how you tried to get around it.

TIPS

You will need to load the following packages for this assessment:

```
library(tidyverse)
library(sp)
library(sf)
library(rnaturalearth)
library(ggrepel)
```

Some Essential Data Wrangling

In the dataset provided for this assessment, each line represents the sale of 1 barrel of whiskey. In order to map the volume of whiskey barrel sales and purchases for each city, you will need to calculate a count of sale and purchase of whiskey barrels per city. For this map, this results in a new data set that looks like this:

place	n	type	lon	lat
Devonport	29	producer	146.3452	-41.1771
Kingston	3	producer	147.3083	-42.9769
Latrobe	1	producer	146.4059	-41.2443
Launceston	77	producer	147.1441	-41.4332
Longford	1	producer	147.1218	-41.5958
Penguin	1	producer	146.0708	-41.1167
Smithton	1	producer	145.1206	-40.8578
Wynyard	1	producer	145.7167	-41.0001
Bridgewater	2	consumer	147.2442	-42.7347
Burnie	12	consumer	145.9052	-41.0529
Devonport	8	consumer	146.3452	-41.1771
Gagebrook	2	consumer	147.2706	-42.7483
Hobart	83	consumer	147.3272	-42.8821
Somerset	6	consumer	145.7876	-41.0574
Ulverstone	1	consumer	146.1872	-41.1703

The following code demonstrates how to re-arrange the data in this way but using a dummy dataset.

```
# create data of flights of 3 staff members of a large organisation
```

```
flights <- tibble::tribble(
  ~departure, ~staff_ID, ~destination,
    "London",    1734,    "Paris",
    "Paris",     1734,    "Barcelona",
    "Paris",     1734,    "Barcelona",
    "Paris",     1734,    "Havana",
    "Paris",     1734,    "Barcelona",
    "Paris",     1734,    "Havana",
    "Paris",     1734,    "Havana",
    "Paris",     1734,    "Havana",
    "Paris",     1734,    "Havana",
    "Paris",     1735,    "Barcelona",
    "Paris",     1734,    "Havana",
    "Paris",     1734,    "Havana",
    "Paris",     1734,    "Havana",
    "Paris",     1734,    "Havana"
```

```

    "Paris",      1738,    "Havana",
    "Paris",      1734,    "Havana",
    "Paris",      1734,    "Havana",
    "Barcelona",  1735,    "Madrid",
    "Paris",      1734,    "London",
    "Barcelona",  1735,    "London",
    "Paris",      1734,    "London",
    "Paris",      1735,    "London",
    "Paris",      1738,    "London",
    "Paris",      1734,    "London",
    "Paris",      1735,    "London",
    "Paris",      1735,    "London",
    "Paris",      1735,    "London",
    "Paris",      1735,    "London",
    "Paris",      1735,    "London",
    "Paris",      1734,    "London",
    "Barcelona",  1734,    "London",
    "Paris",      1734,    "London",
    "Barcelona",  1735,    "London",
    "Paris",      1734,    "London",
    "Paris",      1735,    "London",
    "Paris",      1734,    "London",
    "Paris",      1734,    "London",
    "Barcelona",  1738,    "London",
    "Kingston",   1738,    "Barcelona",
    "Paris",      1734,    "London",
    "Paris",      1738,    "Barcelona",
    "Paris",      1734,    "London",
    "Paris",      1734,    "London",
    "Paris",      1734,    "London",
    "Paris",      1734,    "London",
    "Paris",      1734,    "London",
    "Barcelona",  1738,    "London",
    "Paris",      1734,    "London",
    "Paris",      1734,    "London",
    "Paris",      1734,    "London",
    "Paris",      1738,    "London",
    "Paris",      1734,    "London",
    "Paris",      1734,    "London"
  )

# number of departure flights per city
departures <- flights %>%
  group_by(departure) %>%
  count() %>%
  rename(city = departure) %>%
  add_column(direction = "departure") %>%
  ungroup()

# number of arrival flights per city
destinations <- flights %>%
  group_by(destination) %>%

```

```

count() %>%
rename(city = destination) %>%
add_column(direction = "arrivals") %>%
ungroup()

# Bind the rows of the two data frames
# and change direction column to factor
travel_counts <- rbind(departures, destinations) %>%
  mutate(direction = as_factor(direction))

#load the lat and long of the cities in the tibble.
coordinate <- read.csv(" insert file path here ")

# Join departures and destinations
travel_data <- left_join(travel_counts, coordinates, by = "city")

#Should I remove the date column from the dataset?

```