Week 3

I searched for ideas that we may look into for our project,

* Analysing the survival rate of patients suffering from cancer.
  + Generate synthetic data using synthea for cancer patients in Australia.
  + Standardizing data using OMOP.
  + Visualize using shiny/powerbi/tableau
  + Test using kaplan meier curves
* Comparing the effectiveness of medications/treatments for cancer patients.
* Need to research first which ones to be compared.
* <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2773511/>
* Association between demographic data of a population with cancer disease.
  + Need to find external data (e.g. ABS) for this.
* Impact of insurance on the survival rate of cancer patients <https://onlinelibrary.wiley.com/doi/abs/10.1111/rmir.12138>

Week 4

Websites I found to be useful:

<https://github.com/synthetichealth/synthea>

<https://www.healthit.gov/sites/default/files/page/2022-04/Synthetic%20Health%20Data%20Challenge_Technical%20Guidance%20and%20Tips_508.pdf>

<https://medium.com/@rrossmiller24/my-first-experience-with-synthea-synthetic-health-data-generation-74fffd74a138>

<https://towardsdatascience.com/generating-synthetic-patient-data-b7901c3bd397>

* I cloned the synthea repo from github

*git clone https://github.com/synthetichealth/synthea.git*

*cd synthea*

*./gradlew build check test*

* These commands allowed me to work within this synthea directory, and uses Gradle to build the project, perform checks, and run tests to ensure its correctness and quality. (Gradle is a build automation tool)
* Last command took very long to run (~13 mins) and failed.
* > Task :test4> :test > 240 tests completed, 4 skipped
* Might see this in issue <https://github.com/synthetichealth/synthea/issues/1199>
  + <https://github.com/synthetichealth/synthea/issues/841>
  + <https://github.com/synthetichealth/synthea/pull/935>

*org.mitre.synthea.world.agents.PayerTest > receiveDualEligible FAILEDU3ExporterTest*

*org.junit.ComparisonFailure: expected:<[Dual Eligible]> but was:<[Medicaid]>*

*at org.junit.Assert.assertEquals(Assert.java:117)*

*at org.junit.Assert.assertEquals(Assert.java:146)*

*at org.mitre.synthea.world.agents.PayerTest.receiveDualEligible(PayerTest.java:436)*

*603 tests completed, 1 failed, 7 skipped*

*> Task :test FAILED*

*FAILURE: Build failed with an exception.*

*\* What went wrong:*

*Execution failed for task ':test'.*

*> There were failing tests. See the report at:* [*file:///Users/vaniatjanggra/Documents/GitHub/synthea/synthea/build/reports/tests/test/index.html*](file:///Users/vaniatjanggra/Documents/GitHub/synthea/synthea/build/reports/tests/test/index.html)

*\* Try:*

*> Run with --scan to get full insights.*

*Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.*

*You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.*

*For more on this, please refer to* [*https://docs.gradle.org/8.2.1/userguide/command\_line\_interface.html#sec:command\_line\_warnings*](https://docs.gradle.org/8.2.1/userguide/command_line_interface.html#sec:command_line_warnings) *in the Gradle documentation.*

*BUILD FAILED in 13m 38s*

* Removed the test command so only build and check but still failed (kinda obvious :”))

Basic setup

* Found that using basic setup instructions work.

*cd /Users/vaniatjanggra/downloads*

*(base) Vanias-MacBook-Pro-2:downloads vaniatjanggra$ java -jar synthea-with-dependencies.jar*

* Generating data of 1000 patients and export to a csv file.

*java -jar synthea-with-dependencies.jar -p 1000 -k keep.json --exporter.fhir.export=fa*

*lse --exporter.csv.export=true*

* Need to learn how to generate data according to project and use developer setup so i can customize synthea based on the interest of the project.

Week 5

* Running this code in the terminal.

*git clone https://github.com/synthetichealth/synthea.git*

*cd synthea*

*./run\_synthea*

Output:

*> Task :compileJava*

*Note: Some input files use unchecked or unsafe operations.*

*Note: Recompile with -Xlint:unchecked for details.*

*> Task :run*

*SLF4J: No SLF4J providers were found.*

*SLF4J: Defaulting to no-operation (NOP) logger implementation*

*SLF4J: See http://www.slf4j.org/codes.html#noProviders for further details.*

*Scanned 84 modules and 151 submodules.*

*Loading submodule modules/allergies/allergy\_panel.json*

*Loading submodule modules/allergies/drug\_allergy\_incidence.json*

*Loading submodule modules/allergies/environmental\_allergy\_incidence.json*

*Loading submodule modules/allergies/food\_allergy\_incidence.json*

*Loading submodule modules/allergies/immunotherapy.json*

*Loading submodule modules/allergies/outgrow\_env\_allergies.json*

*....*

*BUILD SUCCESSFUL in 16s*

*4 actionable tasks: 4 executed*

* To specify datas for synthea:

*./run\_synthea -h*   
*> Task :run*  
*Usage: run\_synthea [options] [state [city]]*  
*Options: [-s seed]*  
 *[-cs clinicianSeed]*  
 *[-p populationSize]*  
 *[-r referenceDate as YYYYMMDD]*  
 *[-g gender]*  
 *[-a minAge-maxAge]*  
 *[-o overflowPopulation]*  
 *[-c localConfigFilePath]*  
 *[-d localModulesDirPath]*  
 *[-i initialPopulationSnapshotPath]*  
 *[-u updatedPopulationSnapshotPath]*  
 *[-t updateTimePeriodInDays]*  
 *[-f fixedRecordPath]*  
 *[-k keepMatchingPatientsPath]*  
 *[--config\*=value]*  
 *\* any setting from src/main/resources/synthea.properties*

* For example, i want to generate data of females who have cancer from a population size of 1000.

*./run\_synthea –p 1000 –g F –m \*cancer*

*...*

*997 -- Natosha194 Christopher407 Bayer639 (37 y/o F) Freetown, Massachusetts (39734)*

*1000 -- Norma469 Cole117 (35 y/o F) Boston, Massachusetts (37136)*

*999 -- Debi992 Goldner995 (43 y/o F) Methuen, Massachusetts (45843)*

*996 -- Siobhan540 Dalene805 Harber290 (60 y/o F) West Bridgewater, Massachusetts (64142)*

*995 -- Eloisa55 Isabela97 Paredes726 (79 y/o F) Wayland, Massachusetts (83979)*

*Records: total=1013, alive=1000, dead=13*

*BUILD SUCCESSFUL in 20s*

* However, it still generated other illnesses such as cardiovascular disease. From <https://github.com/synthetichealth/synthea/issues/546> :

*Src/main/java/org/mitre/synthea/engine/Module.java*

Output was:

*bash: src/main/java/org/mitre/synthea/engine/Module.java: Permission denied*

Couldn’t fix this problem.

* Try generating data for Australia (New Zealand) using synthea International as the data generated was Massachussets only

*git clone https://github.com/synthetichealth/synthea-international*

*cd synthea-international*

*cp -R nz/\* ../synthea*

* Failed to work out last command 🙁

*usage: cp [-R [-H | -L | -P]] [-fi | -n] [-apvXc] source\_file target\_file*

*cp [-R [-H | -L | -P]] [-fi | -n] [-apvXc] source\_file ... target\_directory*

* I decide to just work with Massachusetts data from the normal synthea.
* This time I want to output my data into a csv so that I can preprocess it.

*./run\_synthea -p 1000 -g F -m \*cancer --exporter.fhir.export=false --exporter.csv.export=true*

* This command ran, but I couldn’t find the directory in which the csv file is output to.

Week 6

* Tried using the basic setup, despite it taking a longer time to run

*java -jar synthea-with-dependencies.jar -p 2000 -m \*cancer -k keep.json --exporter.fhir.export=false --exporter.csv.export=true*

* Although it produced a warning

*java.lang.RuntimeException: Failed to produce a matching patient after 1000 attempts. Ensure that it is possible for all requested demographics to meet the criteria. (e.g., make sure there is no age restriction that conflicts with a requested condition, such as limiting age to 0-18 and requiring all patients have a condition that only onsets after 55.) If you are confident that the constraints are possible to satisfy but rare, consider increasing the value in config setting `generate.max\_attempts\_to\_keep\_patient`*

*at org.mitre.synthea.engine.Generator.generatePerson(Generator.java:503)*

*at org.mitre.synthea.engine.Generator.lambda$run$3(Generator.java:383)*

*at java.base/java.util.concurrent.FutureTask.run(FutureTask.java:264)*

*at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1136)*

*at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635)*

*at java.base/java.lang.Thread.run(Thread.java:833)*

*Records: total=4519, alive=856, dead=3663*

*RNG=2000*

*Clinician RNG=5647*

* csv exporter outputted these files:

From https://github.com/synthetichealth/synthea/wiki/CSV-File-Data-Dictionary

* Next I want to try to standardize the data I have generated using OMOP

ETL script from: <https://github.com/OHDSI/ETL-Synthea> Note that this is using R script

devtools::[install\_github](https://remotes.r-lib.org/reference/install_github.html)("OHDSI/ETL-Synthea")  
  
 [library](https://rdrr.io/r/base/library.html)(ETLSyntheaBuilder)  
   
cd <- DatabaseConnector::[createConnectionDetails](https://ohdsi.github.io/DatabaseConnector/reference/createConnectionDetails.html)(  
 dbms = "postgresql",   
 server = "localhost/synthea10",   
 user = "postgres",   
 password = "lollipop",   
 port = 5432,   
 pathToDriver = "d:/drivers"   
)  
  
cdmSchema <- "cdm\_synthea10"  
cdmVersion <- "5.4"  
syntheaVersion <- "2.7.0"  
syntheaSchema <- "native"  
syntheaFileLoc <- "/tmp/synthea/output/csv"  
vocabFileLoc <- "/tmp/Vocabulary\_20181119"  
  
ETLSyntheaBuilder::[CreateCDMTables](https://ohdsi.github.io/ETL-Synthea/reference/CreateCDMTables.html)(connectionDetails = cd, cdmSchema = cdmSchema, cdmVersion = cdmVersion)  
   
ETLSyntheaBuilder::[CreateSyntheaTables](https://ohdsi.github.io/ETL-Synthea/reference/CreateSyntheaTables.html)(connectionDetails = cd, syntheaSchema = syntheaSchema, syntheaVersion = syntheaVersion)  
   
ETLSyntheaBuilder::[LoadSyntheaTables](https://ohdsi.github.io/ETL-Synthea/reference/LoadSyntheaTables.html)(connectionDetails = cd, syntheaSchema = syntheaSchema, syntheaFileLoc = syntheaFileLoc)  
   
ETLSyntheaBuilder::[LoadVocabFromCsv](https://ohdsi.github.io/ETL-Synthea/reference/LoadVocabFromCsv.html)(connectionDetails = cd, cdmSchema = cdmSchema, vocabFileLoc = vocabFileLoc)  
   
ETLSyntheaBuilder::[LoadEventTables](https://ohdsi.github.io/ETL-Synthea/reference/LoadEventTables.html)(connectionDetails = cd, cdmSchema = cdmSchema, syntheaSchema = syntheaSchema, cdmVersion = cdmVersion, syntheaVersion = syntheaVersion)

Week 7

* Learning RShiny now,
* Download data from website provided by Siavash <https://portal.gdc.cancer.gov/>
* Three different types of file: biospecimen, clinical, manifest, all in JSON format.
* To load JSON file to R

*install.packages("rjson")*

From <https://makemeanalyst.com/r-programming/reading-and-writing-data-to-and-from-r/json-files-in-r/#:~:text=To%20get%20JSON%20files%20into,to%20read%20the%20json%20file>.

*library(shiny)*  
*library(rjson)*  
*mydata <- fromJSON(file=”...”)*

From <https://stackoverflow.com/questions/56226664/uploading-json-files-using-fileinput-shiny-app>

*library(shiny)*  
*library(RJSONIO)*  
*library(rjson)*  
*library(DT)*  
*write(toJSON(mtcars), "test.json")*  
  
*shinyApp(*  
 *ui = fluidPage(*  
 *fileInput("Json", "Choose Json File",*  
 *multiple = FALSE,*  
 *accept = c(".json")),*  
 *DTOutput('tbl')*  
 *),*  
 *server = function(input, output) {*  
 *output$tbl = renderDT({*  
 *req(input$Json)*  
 *as.data.frame(fromJSON(file = input$Json$datapath))*  
 *})*  
 *}*  
*)*

* Need to work on operating RShiny to get the information we need

Week 8

* Preprocessed the patients.csv data I generated from week 6.
  + Added a column “died” indicating they have died.
  + Allocated random suburbs to each patient (as the data produced is in masachusetts)

import **pandas** as **pd**

import **random**

import **numpy** as **np**

*# add another column to show whether patient is still alive or dead*

patients = **pd**.**read\_csv**('patients.csv')

patients['STATUS'] = **np**.**where**(patients['DEATHDATE'].**isnull**(), 'Alive', 'Dead')

columns\_to\_remove = ['DRIVERS', 'SUFFIX', 'COUNTY', 'BIRTHPLACE', 'ADDRESS', 'CITY', 'STATE', 'RACE', 'ETHNICITY', 'FIPS', 'ZIP', 'LAT', 'LON']

*# Drop the specified columns*

patients = patients.**drop**(columns=columns\_to\_remove)

*# create new column to allocate each patient to a random Melbourne Suburb*

suburbs = ["Carlton", "Carlton North", "Docklands", "East Melbourne",

"Flemington", "Kensington", "Melbourne", "North Melbourne",

"Parkville", "Port Melbourne", "Southbank", "South Wharf",

"South Yarra", "West Melbourne", "Albert Park", "Balaclava",

"Elwood", "Middle Park", "Ripponlea", "St Kilda", "St Kilda East",

"St Kilda West", "South Melbourne", "Abbotsford", "Alphington",

"Burnley", "Clifton Hill", "Collingwood", "Cremorne", "Fairfield",

"Fitzroy", "Fitzroy North", "Princes Hill", "Richmond"]

states = ["Victoria"]

patients['SUBURB'] = patients.**apply**(lambda x: **random**.choice(suburbs), axis=1)

patients['STATE'] = patients.**apply**(lambda x: **random**.choice(states), axis=1)

patients.**head**(20)

output\_path = '.../patients.csv'

*# Save the modified DataFrame to a new CSV file*

patients.**to\_csv**(output\_path, index=False)

* Note that this code is a modified version of Xuan Hung’s code (<https://github.com/Clinical-Informatics-Collaborative/socio-economic-data/blob/main/Shiny%20Dashboard/generate-fake-data.ipynb>)
* Now that the data is preprocessed, visualize it in RStudio.
* Installing required packages
  + Unable to download ‘sf’ package to read shapefiles?
  + When installed, cannot load the library

*> shiny::runApp('Desktop/wehi/shiny')*

*Loading required package: shiny*

*Attaching package: ‘dplyr’*

*The following objects are masked from ‘package:stats’:*

*filter, lag*

*The following objects are masked from ‘package:base’:*

*intersect, setdiff, setequal, union*

*Error in library(sf) : there is no package called ‘sf’*

* Tried answering ‘No’ when asked ‘do you want to install from sources the package which needs compilation?’ <https://community.rstudio.com/t/library-wont-load-after-installation/59342>
* Still did not work.
* Tried restarting R and updating and restart
* <https://community.rstudio.com/t/how-to-fix-sf-installation-issue/154085/9> this link helped me fix it, i reinstalled the latest version of R and restarted RStudio, works
* Visualized the survival rate of lung cancer patients per suburb.

*#*

*# 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/*

*#*

*# Load libraries*

*library(shiny)*

*library(dplyr)*

*library(ggplot2)*

*library(sf)*

*library(leaflet)*

*# Read lung cancer patient data*

*lung\_cancer\_data <- read.csv("patients.csv")*

*# Perform data manipulations to calculate survival rate*

*lung\_cancer\_data$DEATHDATE <- as.Date(lung\_cancer\_data$DEATHDATE)*

*lung\_cancer\_data$BIRTHDATE <- as.Date(lung\_cancer\_data$BIRTHDATE)*

*lung\_cancer\_data$SurvivalRate <- ifelse(is.na(lung\_cancer\_data$DEATHDATE), "Alive", "Dead")*

*# Create Shiny app*

*shinyApp(*

*ui = fluidPage(*

*titlePanel("Survival Rate of Lung Cancer Patients by Suburb"),*

*mainPanel(*

*plotOutput("survivalPlot")*

*)*

*),*

*server = function(input, output, session) {*

*output$survivalPlot <- renderPlot({*

*suburb\_survival\_data <- lung\_cancer\_data %>%*

*group\_by(Suburb) %>%*

*summarise(*

*SurvivalRate = sum(is.na(DEATHDATE)) / n() # Alive / Total Patients*

*)*

*ggplot(suburb\_survival\_data, aes(x = Suburb, y = SurvivalRate)) +*

*geom\_bar(stat = "identity", fill = "skyblue") +*

*labs(title = "Survival Rate of Lung Cancer Patients by Suburb",*

*x = "Suburb",*

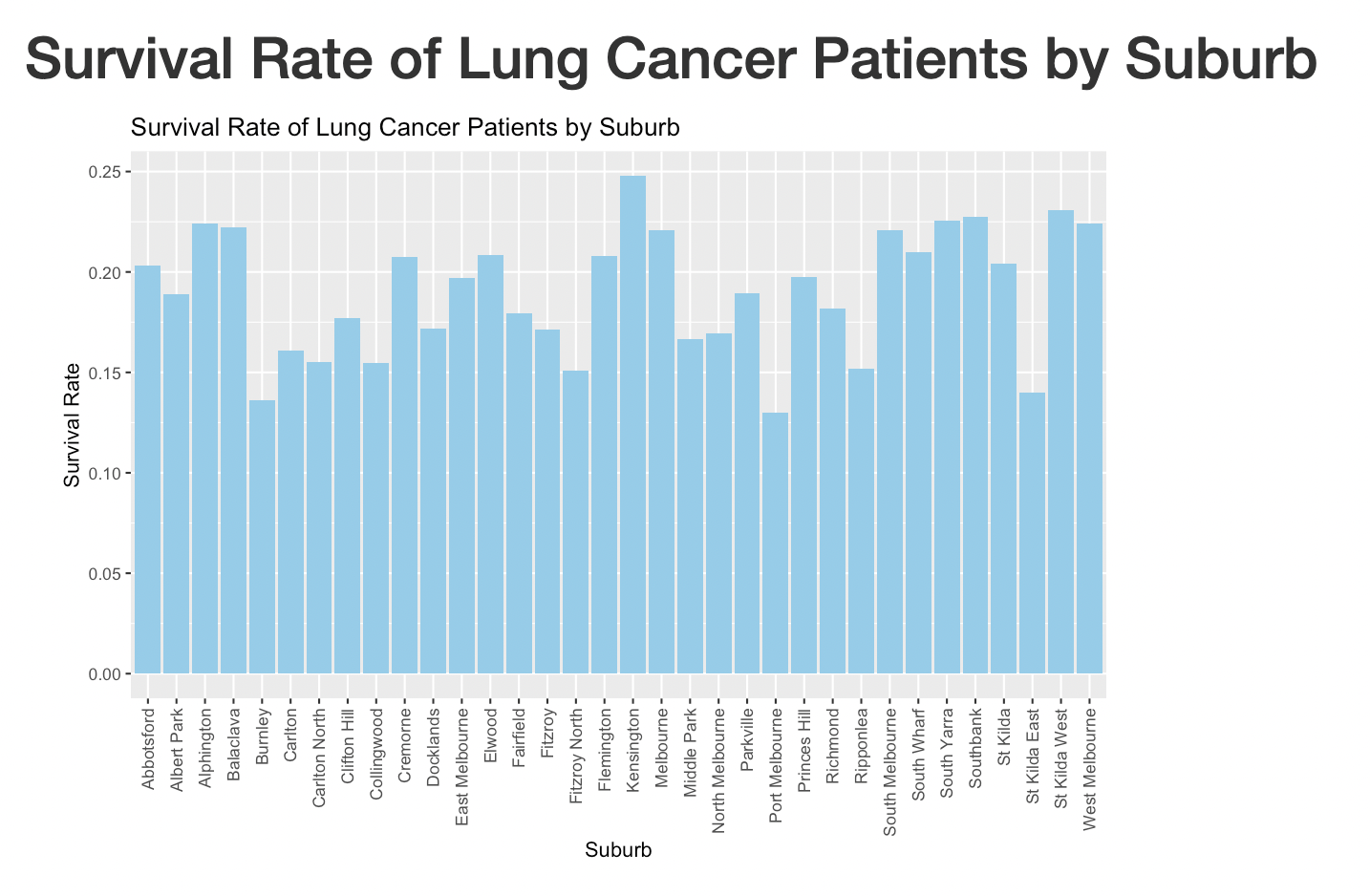
*y = "Survival Rate") +*

*theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust = 1))*

*})*

*}*

*)*



* Plotted against gender

# Create Shiny app

shinyApp(

ui = fluidPage(

titlePanel("Survival Rate of Lung Cancer Patients by Gender"),

mainPanel(

plotOutput("survivalPlot")

)

),

server = function(input, output, session) {

output$survivalPlot <- renderPlot({

gender\_survival\_data <- lung\_cancer\_data %>%

group\_by(GENDER) %>%

summarise(

SurvivalRate = sum(is.na(DEATHDATE)) / n() # Alive / Total Patients

)

ggplot(gender\_survival\_data, aes(x = GENDER, y = SurvivalRate, fill = GENDER)) +

geom\_bar(stat = "identity", position = "dodge", color = "black") +

labs(title = "Survival Rate of Lung Cancer Patients by Gender",

x = "Gender",

y = "Survival Rate") +

scale\_fill\_manual(values = c("M" = "skyblue", "F" = "pink")) +

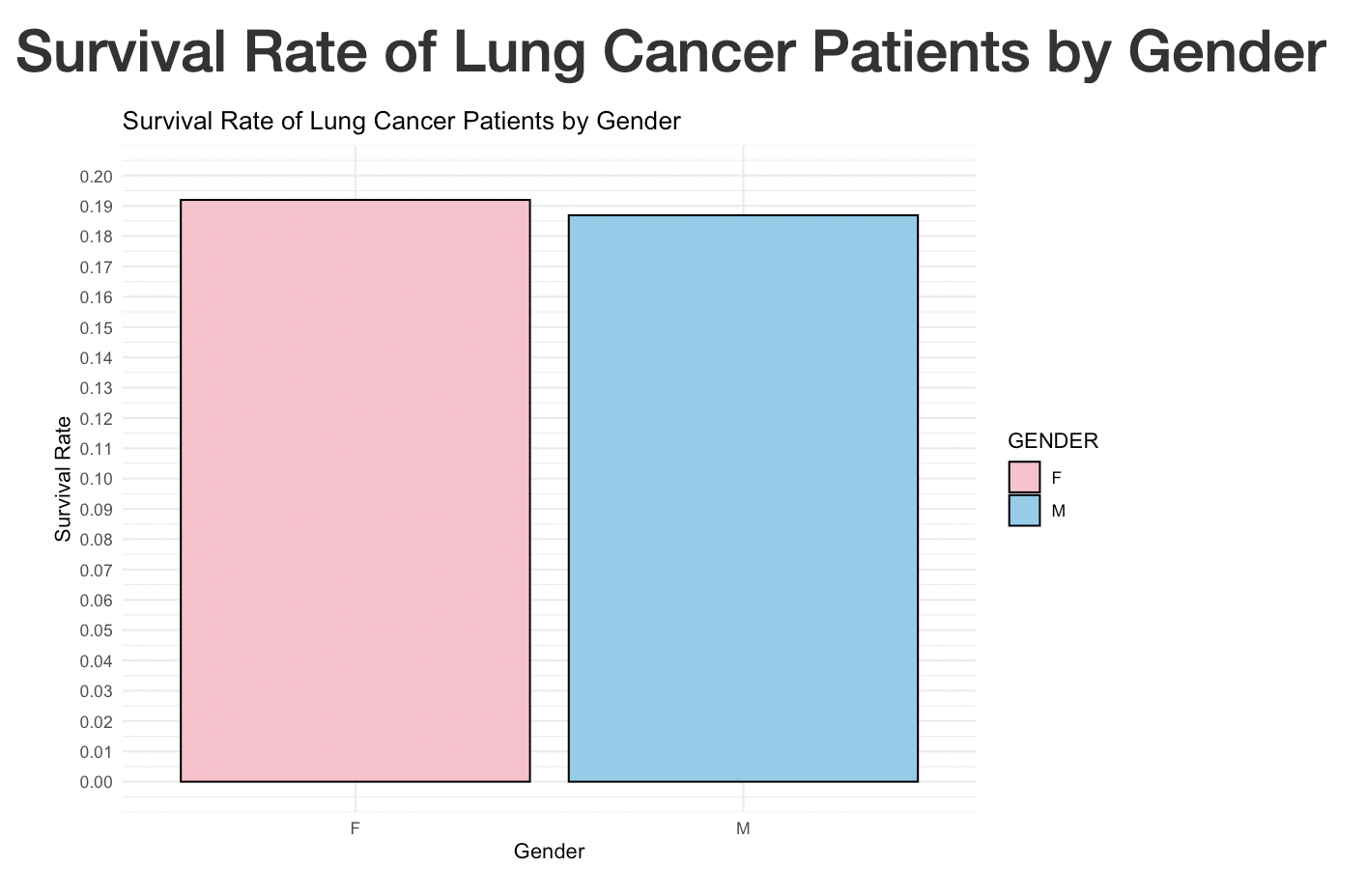
theme\_minimal() +

scale\_y\_continuous(limits = c(0, 0.2), breaks = seq(0, 0.2, by = 0.01))

})

}

)



* I then used Xuan Hung’s code for producing heat maps for each suburb in a map that is user interactive.

Listening on http://127.0.0.1:4531

Warning: sf layer has inconsistent datum (+proj=longlat +ellps=GRS80 +no\_defs).

Need '+proj=longlat +datum=WGS84'

Warning in RColorBrewer::brewer.pal(max(3, n), palette) :

n too large, allowed maximum for palette YlOrRd is 9

Returning the palette you asked for with that many colors

Warning in RColorBrewer::brewer.pal(max(3, n), palette) :

n too large, allowed maximum for palette YlOrRd is 9

Returning the palette you asked for with that many colors

Warning: Error in dplyr::group\_by: Must group by variables found in `.data`.

Column `ETHNICITY` is not found.

Column `VALUE` is not found.

* Modified my python code to not drop ethnicity and race from the columns, and change income range.

suburbs = ["Carlton", "Carlton North", "Docklands", "East Melbourne",

"Flemington", "Kensington", "Melbourne", "North Melbourne",

"Parkville", "Port Melbourne", "Southbank", "South Wharf",

"South Yarra", "West Melbourne", "Albert Park", "Balaclava",

"Elwood", "Middle Park", "Ripponlea", "St Kilda", "St Kilda East",

"St Kilda West", "South Melbourne", "Abbotsford", "Alphington",

"Burnley", "Clifton Hill", "Collingwood", "Cremorne", "Fairfield",

"Fitzroy", "Fitzroy North", "Princes Hill", "Richmond"]

postal\_codes = ['3000', '3004', '3008', '3010', '3015', '3025']

states = ["Victoria"]

income\_ranges = ['<30,000', '30,000-60,000', '60,000-90,000', '90,000-120,000', '>120,000']

weights=[0.6, 0.05, 0.15, 0.15, 0.05]

patients['Suburb'] = patients.**apply**(lambda x: **random**.choice(suburbs), axis=1)

patients['Postal Code'] = patients.**apply**(lambda x: **random**.choice(postal\_codes), axis=1)

patients['State'] = patients.**apply**(lambda x: **random**.choice(states), axis=1)

patients['INCOME'] = patients.**apply**(lambda x: **random**.choices(income\_ranges, weights)[0]

if x['Suburb'] in ["Docklands", "Melbourne"] else **random**.choice(income\_ranges),

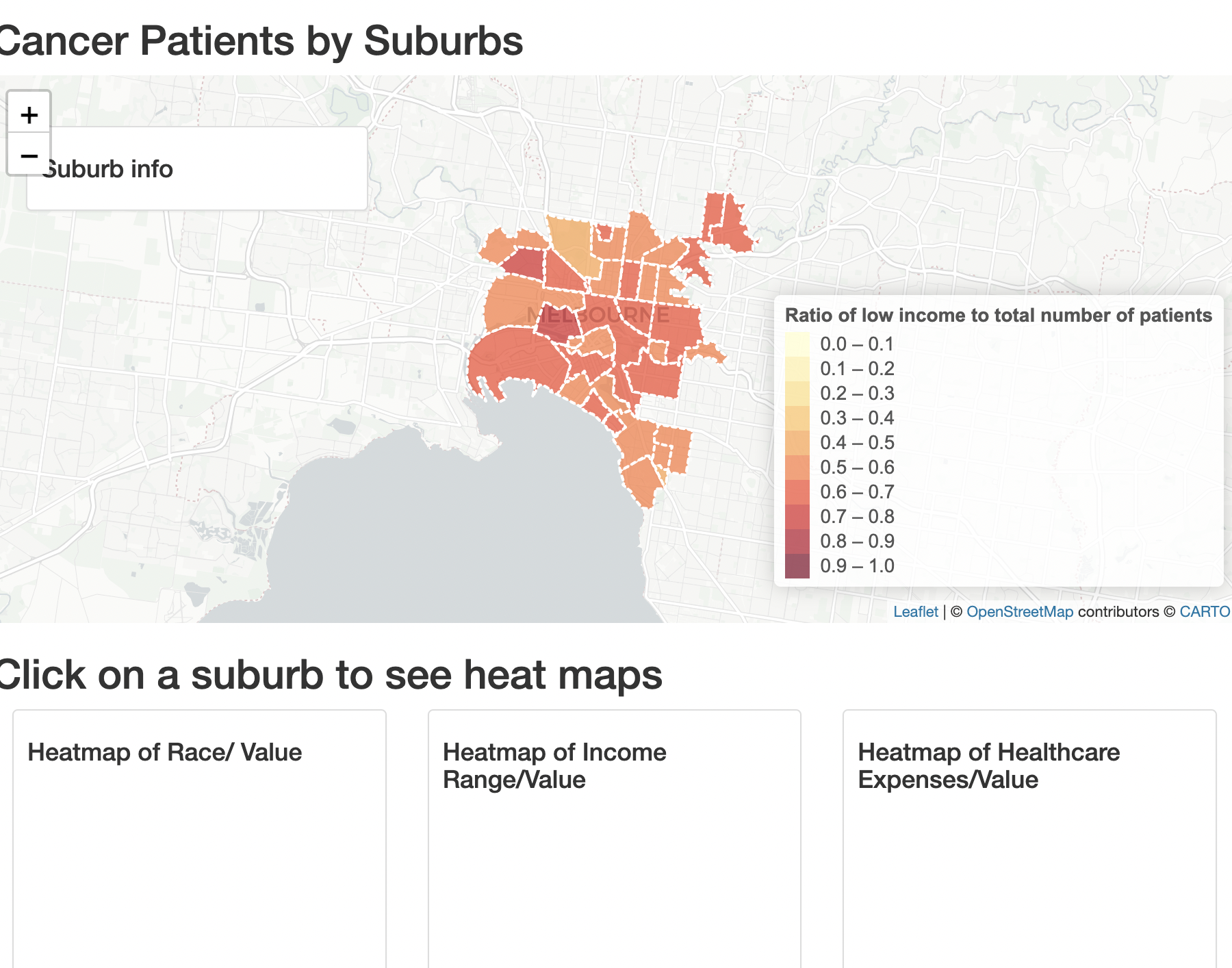
axis=1)

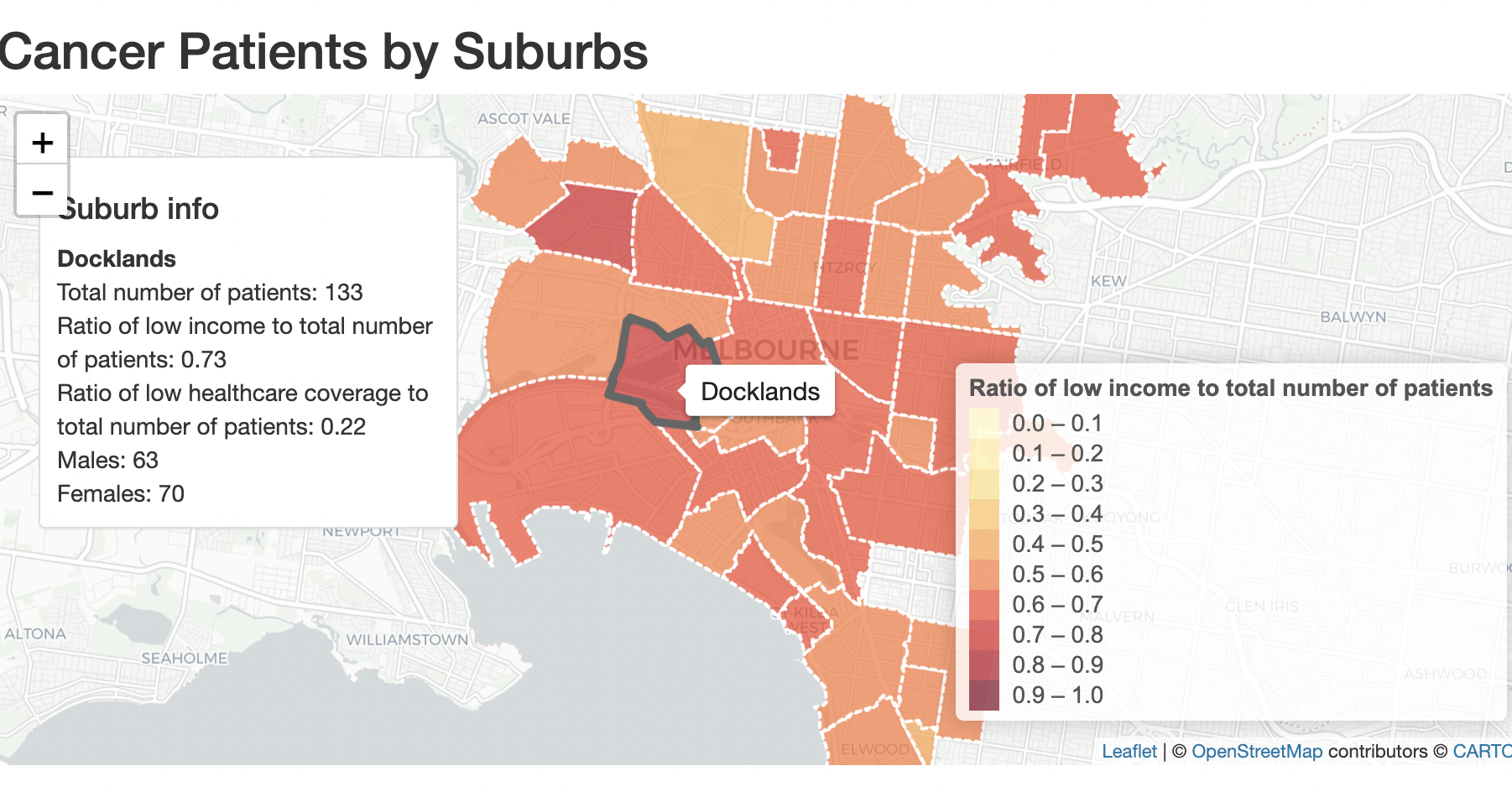
* Used code from Xuan Hung, modified to display ratio of low healthcare coverage in suburb info. Ui <- <https://github.com/Clinical-Informatics-Collaborative/socio-economic-data/blob/main/Shiny%20Dashboard/dashboard/ui.R>

Server <- <https://github.com/Clinical-Informatics-Collaborative/socio-economic-data/blob/main/Shiny%20Dashboard/dashboard/server.R>

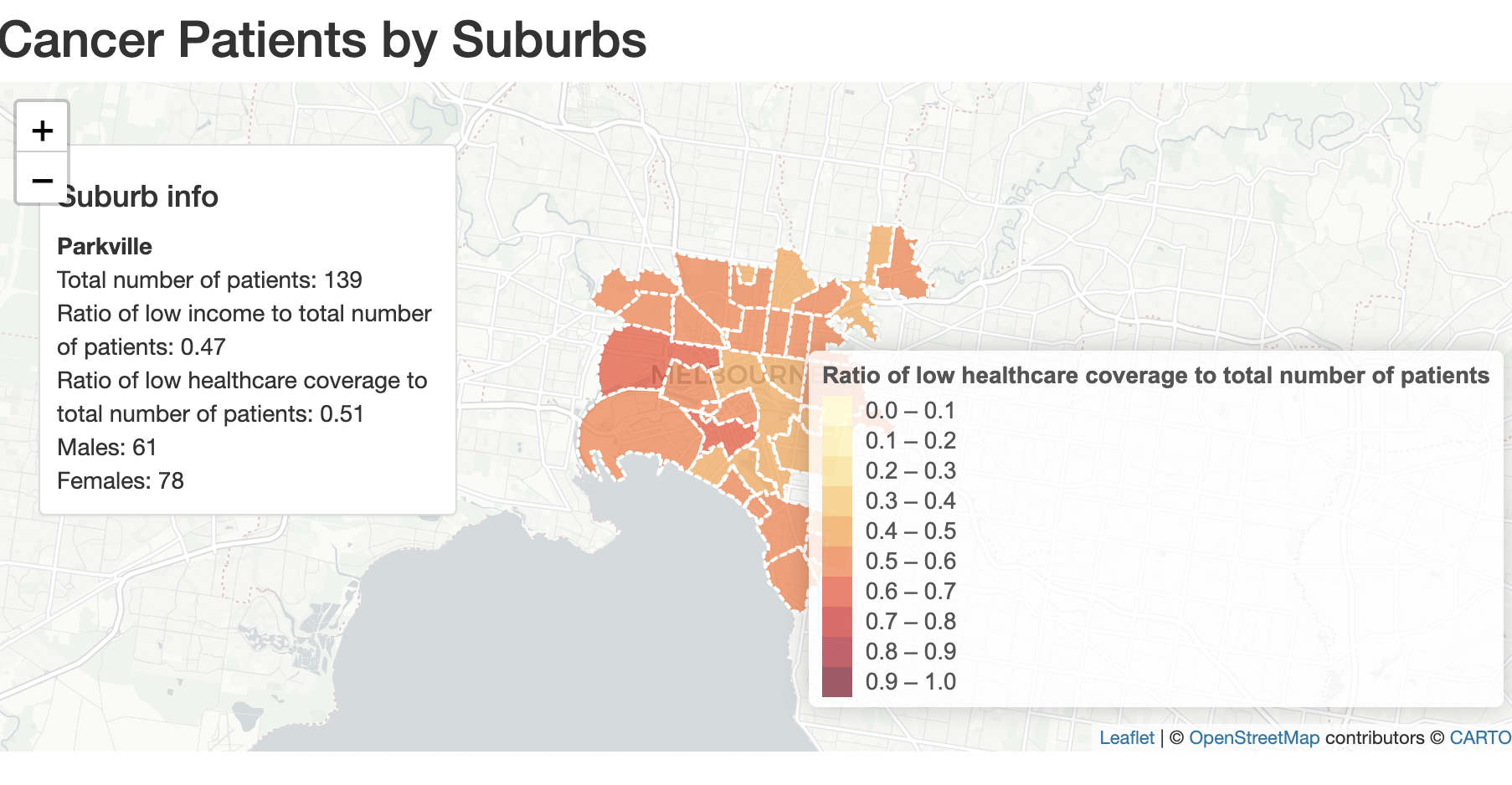
From server, i added

ratio\_healthcare = sum(HEALTHCARE\_COVERAGE < 60000) / total\_patients





* Also generated the heatmap for low healthcare coverage with threshold of 110,000 as it produced the most diverse heatmap



* However, i couldn’t work out how to display the “click suburb for heatmap” options, it only closes the viewer and stays running but does not do anything.