ResBaz2023

Anthony Kimpton

22 November, 2023

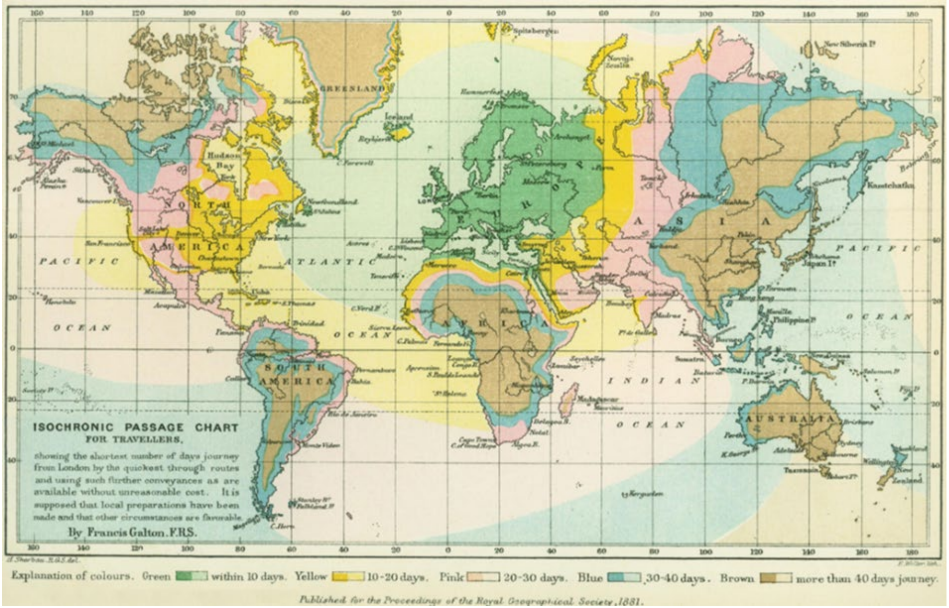
GitHub Repo containing this script

GitHub Repo

Chrono-urbanism?

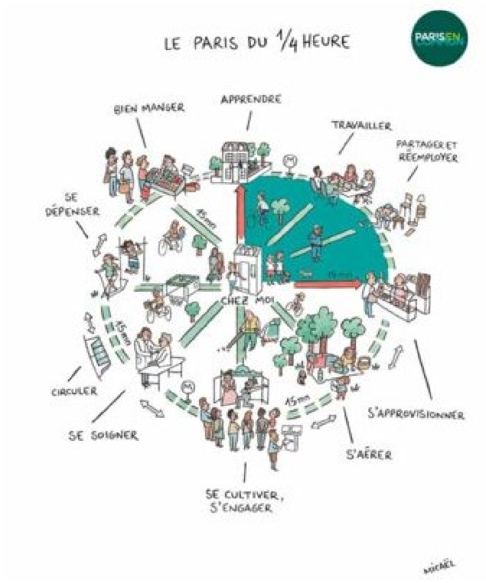
- “Mobility” operationalised as *time* rather than *distance*
- “Humane” and “Equitable” if where we live should not impact opportunities
- “Marchetti’s Constant” that 1 hour of daily travel is an anthropological invariant

Early Example of this thinking


ISOCHRONIC PASSAGE CHART
FOR TRAVELLERS,
showing the shortest number of days journey from London by the quickest through routes and using such further conveyances as are available without unreasonable cost. It is supposed that local preparations have been made and that other circumstances are favorable.
By Francis Galton.F.R.S.
NORTH AMERICA
SOUTH AMERICA
EUROPE
AFRICA
ASIA
AUSTRALIA
GREENLAND
PACIFIC OCEAN
ATLANTIC OCEAN
INDIAN OCEAN
PACIFIC OCEAN
Explanation of colours. Green 🟩 within 10 days. Yellow 🟨 10-20 days. Pink ▭ 20-30 days. Blue 🟦 30-40 days. Brown 🟫 more than 40 days journey.
Published for the Proceedings of the Royal Geographical Society, 1881.

Galton F. (1881) On the construction of isochronic passage-charts. *Proceedings of the Royal Geographical Society and Monthly Record of Geography

x-minute cities/neighbourhoods



15 minute cities

x-minute cities/neighbourhoods

30-minute city aims:

- the Australian Federal Government’s *2016 Smart Cities Plan*
- New South Wales State Government’s *2014 A Plan for Growing Sydney*
- Queensland State Government’s *2019 Queensland Transport Strategy*

20-minute neighbourhoods:

- Victorian Government’ *2017 Plan Melbourne*

20-minute city & 10-minute neighbourhood:

- the Ipswich City Council’s (ICC) *iGO: City of Ipswich Transport Plan*

Multimodal routing with R’s r5r

- homepage

Setting up: the usual suspects

```
#install.packages(c("RColorBrewer",
"viridis", "tidyverse", "sf",
"data.table", "ggplot2", "interp",
"tidytransit","DiagrammeR"…))
library(RColorBrewer)
library(viridis)
library(tidyverse)
library(sf)
library(data.table)
library(interp)
library(tidytransit)
library(DiagrammeR)
#
remotes::install_github("bergant/datamo
delr")
library(datamodelr)
library(elevatr)
library(stars)
```

```
library(gtfstools)
library(tmap)
library(reactable)
library(plotly)

wd <- getwd()
date <- format(Sys.time(), '%y%m%d')

download_dir <- file.path(wd,
paste("downloads", date, sep = "_"))

if (!file.exists(download_dir)){
  dir.create(download_dir)
}

tmap_mode("view")
tmap_mode("plot")
tmap_options(check.and.fix = TRUE)
```

Setting up: the problem pair

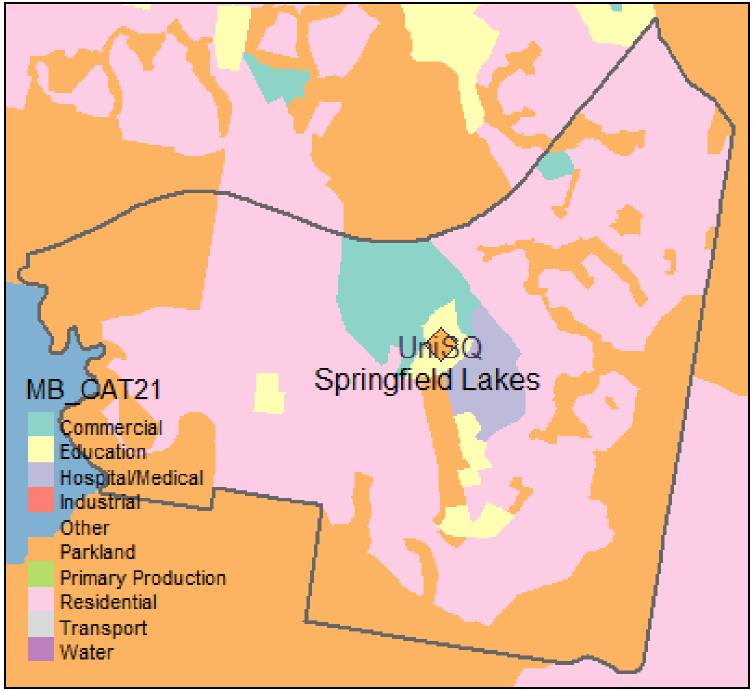
```r
#If it wont't run use this R version that has been compiled differently
https://cran.r-project.org/bin/windows/base/rpatched.html
#This must run before loading rJava
https://stackoverflow.com/questions/34624002/r-error-java-lang-outofmemoryerror-java-heap-space
#install.packages(c("rJava", "r5r"))
library(rJava)
library(r5r)

options(java.parameters = "-Xmx8000m")

.jinit()
ver <- .jcall("java.lang.System", "S", "getProperty", "java.version")
ver <- as.numeric(gsub("\\..*", "", ver))
ver
```

Areas: Downloads

```r
#local government areas
url_lga_2021 <-
"https://www.abs.gov.au/statistics/standards/australian-statistical-
geography-standard-asgs-edition-3/jul2021-jun2026/access-and-
downloads/digital-boundary-files/LGA_2021_AUST_GDA2020_SHP.zip"

download.file(url_lga_2021, destfile = paste0(download_dir, "/",
basename(url_lga_2021)))

#Suburbs
url_sa2_2021 <-
"https://www.abs.gov.au/statistics/standards/australian-statistical-
geography-standard-asgs-edition-3/jul2021-jun2026/access-and-
downloads/digital-boundary-files/SA2_2021_AUST_SHP_GDA2020.zip"

download.file(url_sa2_2021, destfile = paste0(download_dir, "/",
basename(url_sa2_2021)))

#land use areas
url_meshblocks_2021 <-
"https://www.abs.gov.au/statistics/standards/australian-statistical-
geography-standard-asgs-edition-3/jul2021-jun2026/access-and-
downloads/digital-boundary-files/MB_2021_AUST_SHP_GDA2020.zip"

download.file(url_meshblocks_2021, destfile = paste0(download_dir,
"/", basename(url_meshblocks_2021)))

for (file in list.files(path = download_dir, pattern="*.zip")){
  print(file)
  unzip(zipfile = paste(download_dir, file, sep = "/"), exdir =
download_dir, overwrite = TRUE)
}
```

```r
for (file in list.files(path = download_dir, pattern="*.shp$")){
  print(file)
  assign(gsub(".shp", "",file), st_read(paste(download_dir, file,
sep = "/")))
}

SA2_2021_AUST_GDA2020 <- SA2_2021_AUST_GDA2020 %>%
  filter(GCC_NAME21 == "Greater Brisbane")

study_area <- SA2_2021_AUST_GDA2020 %>%
  filter(SA2_NAME21 == "Springfield Lakes")

land_use <- MB_2021_AUST_GDA2020 %>%
  filter(GCC_NAME21 == "Greater Brisbane")

point_of_origin_sf <- land_use %>%
  filter(MB_CODE21 == 30562523100) %>%
  st_centroid() %>%
  st_transform(crs = "WGS84") %>%
  transmute(id = "UniSQ")

point_of_origin_df <- point_of_origin_sf %>%
  data.frame(st_coordinates(st_cast(.$geometry,"MULTIPOINT"))) %>%
  st_drop_geometry() %>%
  transmute(id, lon = X, lat = Y)

origins_and_destinations_df <- SA2_2021_AUST_GDA2020 %>%
  st_centroid()  %>%
  st_transform(crs = "WGS84") %>%
  data.frame(st_coordinates(st_cast(.$geometry,"MULTIPOINT"))) %>%
  st_drop_geometry() %>%
  transmute(id = SA2_NAME21, lon = X, lat = Y)
```

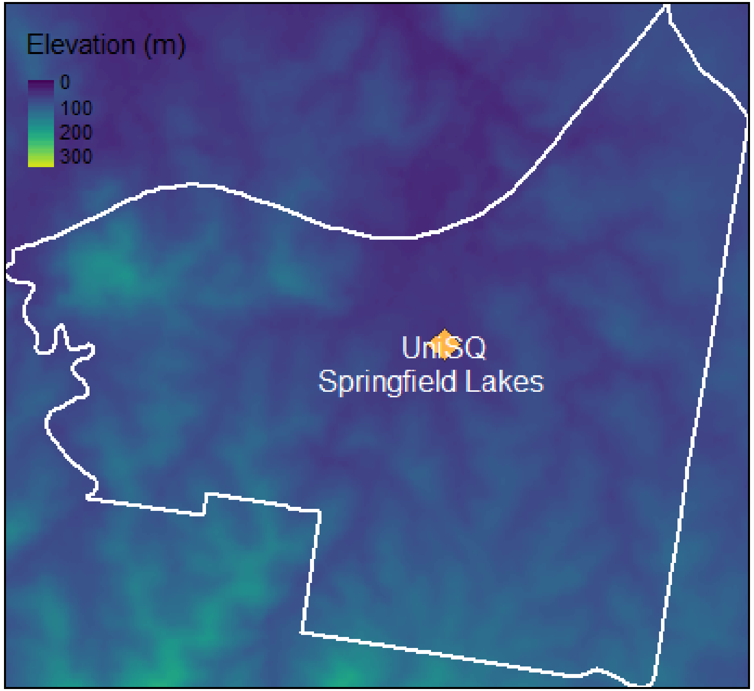
Areas: Mapped



Elevation: Download

```
options(timeout = 8000) #for when it's a biggie or
slow

elevation <- get_elev_raster(study_area, z = 12)
%>% st_as_stars()

write_stars(elevation, paste0(download_dir,
"/elevation.tif"))

#contours <- st_contour(elevation)

options(timeout = 60) #back to 1 min default
```

Elevation: Mapped

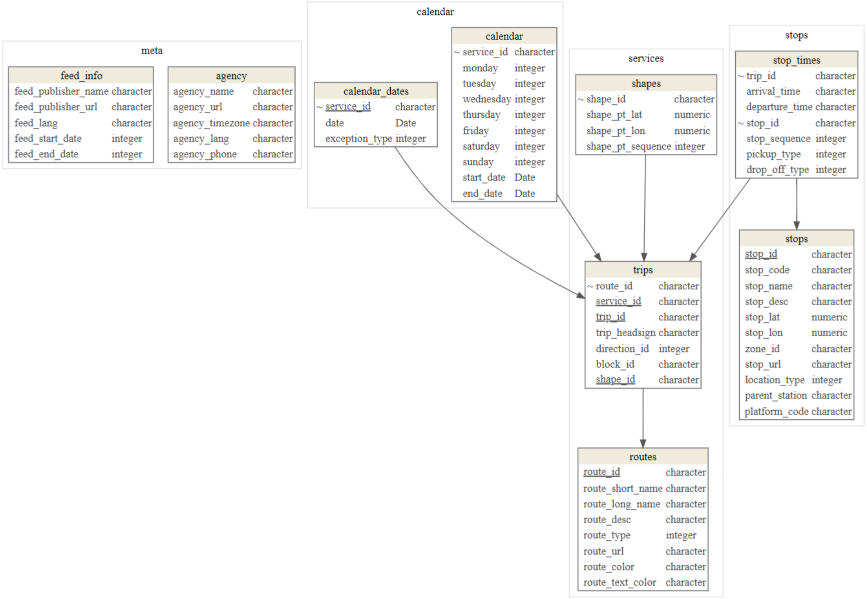


GTFS: Download

- OpenMobilityData

```
# Public Transport Routes, Stops, and Schedules
url_latest_gtfs <- "https://gtfsrt.api.translink.com.au/GTFS/SEQ_GTFS.zip"

download.file(url_latest_gtfs, destfile = paste0(download_dir, "/",
basename(url_latest_gtfs)))

latest_validator <- download_validator(tempdir())

gtfs <- read_gtfs(paste0(download_dir, "/SEQ_GTFS.zip"))

path_output_dir <- tempfile("validation_from_path")

validate_gtfs(gtfs, ".", latest_validator)

feed_info <- read.csv(paste(download_dir, "feed_info.txt", sep = "/"))
```

GTFS: Datamodel



meta

feed_info	
feed_publisher_name	character
feed_publisher_url	character
feed_lang	character
feed_start_date	integer
feed_end_date	integer

agency	
agency_name	character
agency_url	character
agency_timezone	character
agency_lang	character
agency_phone	character

calendar

calendar_dates	
~ service_id	character
date	Date
exception_type	integer

calendar	
~ service_id	character
monday	integer
tuesday	integer
wednesday	integer
thursday	integer
friday	integer
saturday	integer
sunday	integer
start_date	Date
end_date	Date

services

shapes	
~ shape_id	character
shape_pt_lat	numeric
shape_pt_lon	numeric
shape_pt_sequence	integer

trips	
~ route_id	character
service_id	character
trip_id	character
trip_headsign	character
direction_id	integer
block_id	character
shape_id	character

routes	
route_id	character
route_short_name	character
route_long_name	character
route_desc	character
route_type	integer
route_url	character
route_color	character
route_text_color	character

stops

stop_times	
~ trip_id	character
arrival_time	character
departure_time	character
~ stop_id	character
stop_sequence	integer
pickup_type	integer
drop_off_type	integer

stops	
stop_id	character
stop_code	character
stop_name	character
stop_desc	character
stop_lat	numeric
stop_lon	numeric
zone_id	character
stop_url	character
location_type	integer
parent_station	character
platform_code	character

Brisbane’s GTFS Data Model

OSM: Download

- *https://download.geofabrik.de/australia-oceania/australia.html*
- *https://download.bbbike.org/osm/bbbike/Brisbane/*

```
# Roads and Points of Interest
#Australian OSM will throw out an unsolved error that most likely relates to Java'
tricky memory management
#url_latest_osm <- "https://download.geofabrik.de/australia-oceania/australia-
latest.osm.pbf"

url_latest_osm <-
"https://download.bbbike.org/osm/bbbike/Brisbane/Brisbane.osm.pbf"

options(timeout = 8000) #for when it's a biggie or slow

download.file(url_latest_osm, destfile = paste0(download_dir, "/",
basename(url_latest_osm)))

options(timeout = 60) #back to 1 min default
```

Over to the script