

## rainovertime\_2019

```
library(dplyr)
```

```
##
```

```
## 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
```

```
library(tmap)
```

```
library(ggplot2)
```

```
library(cowplot)
```

```
library(magick)
```

```
## Linking to ImageMagick 6.9.12.3
```

```
## Enabled features: cairo, fontconfig, freetype, heic, lcms, pango, raw, rsvg, webp
```

```
## Disabled features: fftw, ghostscript, x11
```

```
library(sf)
```

```
## Linking to GEOS 3.9.1, GDAL 3.4.0, PROJ 8.1.1; sf_use_s2() is TRUE
```

```
library(readr)
```

```
library(tidyr)
```

```
## Uploading rainfall data and the corresponding lat/long locations from another data file
```

```
#this is pulling in the rainfall data for 2019, information collected every 15min
```

```
rainfall_data <- read_csv("/Users/Tyrah/adv GIS classwork/finalproject_687/BaltCity2019_Tyrah_finalproj
```

```
## Rows: 17464 Columns: 244
```

```
## -- Column specification -----
```

```
## Delimiter: ","
```

```
## dbl (242): 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 2...
```

```
## date (1): Date
```

```
## time (1): Time
```

```
##
```

```
## i Use 'spec()' to retrieve the full column specification for this data.
```

```
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
#this is the location of the pixels collecting the rainfall data
pixels_latlong <- read_csv("/Users/Tyrah/adv GIS classwork/finalproject_687/Balt_latlong.csv")
```

```
## Rows: 242 Columns: 3
## -- Column specification -----
## Delimiter: ","
## dbl (3): PixelNumber, latitude, longitude
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
# this is getting the geometries of the pixels
# pixel_location <- st_as_sf(pixels_latlong, coords = c("longitude", "latitude"), crs = 4326)
# plot(pixel_location$geometry)
```

```
rainfall_data_transposed <- rainfall_data %>% pivot_longer(cols = 3:244, names_to = "gridnum", values_to = "rainmm")
```

```
#had to change this column to be of character type (original a double), so that it could join with the pixels_latlong
pixels_latlong$PixelNumber <- as.character(pixels_latlong$PixelNumber)
```

```
rainwithlatlong <- left_join(rainfall_data_transposed, pixels_latlong, by = c("gridnum" = "PixelNumber"))
```

```
rainwithlatlong$gridnum <- as.vector(rainwithlatlong$gridnum)
```

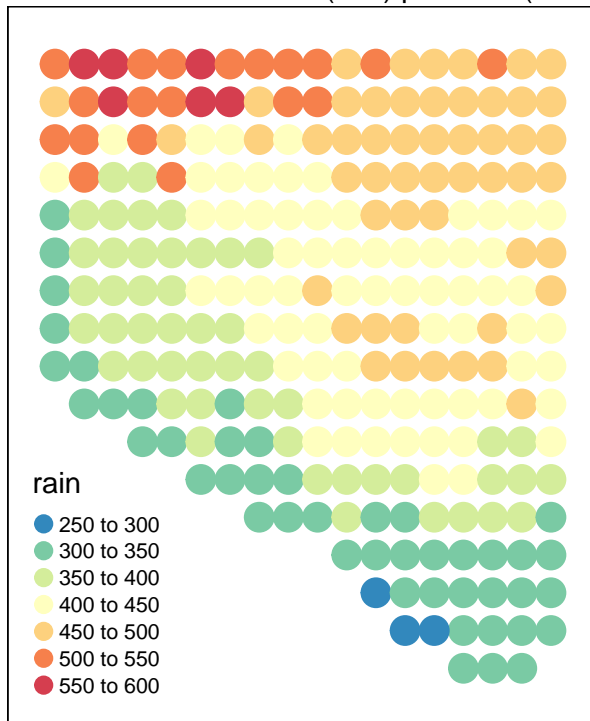
```
#if i dont use the removed zeros in the group by can also use rainwithlatlong
removed_zeros <- filter(rainwithlatlong, rainmm > 0)
dayandpixel <- removed_zeros %>% group_by(Date, gridnum, latitude, longitude)
# sumbyday_rain <- dayandpixel %>% summarise(
#   sum = sum(rainmm)
# )
```

```
by_gridnum <- rainwithlatlong %>% group_by(gridnum, latitude, longitude)
sum_rain <- by_gridnum %>% summarise(
  rain = sum(rainmm),
  mean = mean(rainmm)
)
```

```
rain_sf <- st_as_sf(sum_rain, coords = c("longitude", "latitude"), crs = 4326)
rain_proj <- rain_sf %>% st_transform(3857)
```

```
totalrain_plot <- tm_shape(rain_sf) +
  tm_dots(group = "rain", col = "rain", size = 1, palette = "-Spectral") +
  tm_layout( outer.margins = rep(0.06, 6), inner.margins = rep(0.08, 8), main.title = "Total Amount of Rainfall")
```

Total Amount of Rainfall(mm) per Pixel (2019)



```
# https://stackoverflow.com/questions/28335715/r-how-to-filter-subset-a-sequence-of-dates
# Create date object
dayandpixel$Date <- as.Date(dayandpixel$Date)

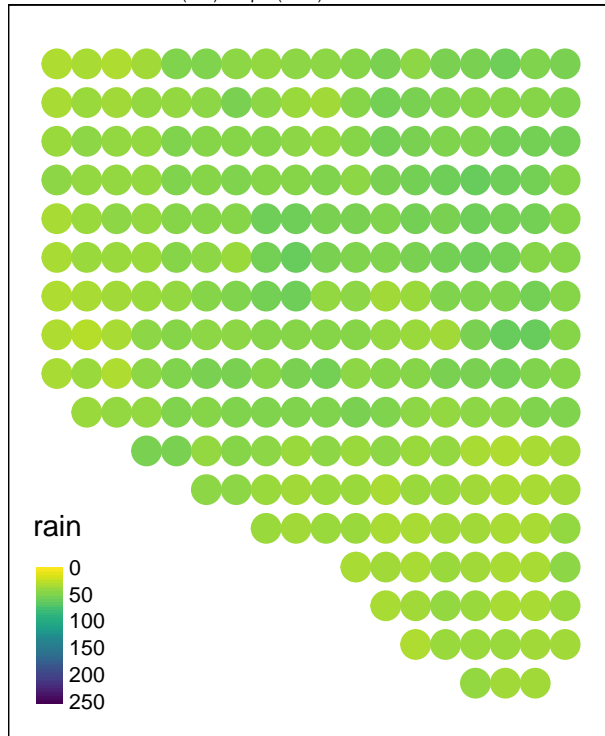
april <- filter(dayandpixel, between(Date, as.Date("2019-04-01"), as.Date("2019-05-1")))
april_rain <- april %>% group_by(gridnum, latitude, longitude)
totalapril <- april_rain%>% summarise(
  rain = sum(rainmm))
```

```
## 'summarise()' has grouped output by 'gridnum', 'latitude'. You can override
## using the '.groups' argument.
```

```
april_sf <- st_as_sf(totalapril, coords = c("longitude", "latitude"), crs = 4326)
```

```
april_rain <- tm_shape(april_sf) +
  tm_dots(group = "rain", col = "rain", size = 1, palette = "-viridis", style="cont", breaks = c(0, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600))
  tm_layout( outer.margins = rep(0.06, 6), inner.margins = rep(0.08, 8), main.title = "Total Amount of Rainfall in April 2019")
april_rain
```

Total Amount of Rainfall(mm) in April (2019)



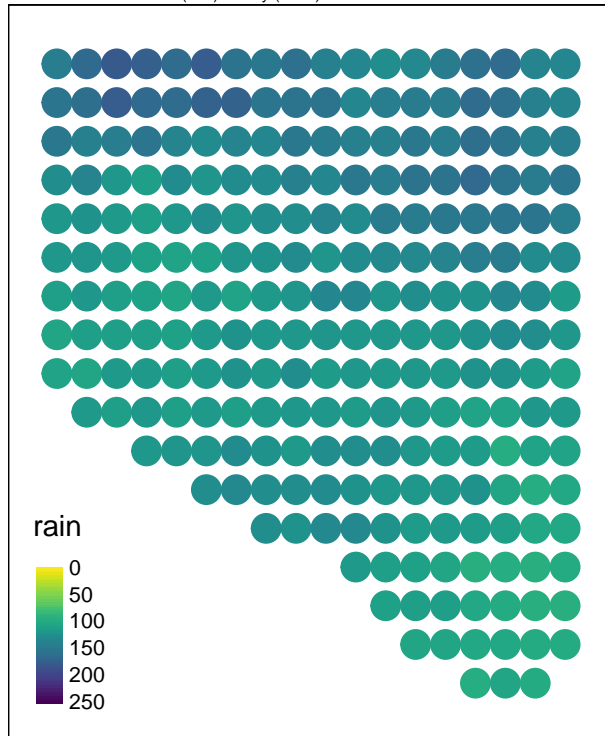
```
may <- filter(dayandpixel, between(Date, as.Date("2019-05-01"), as.Date("2019-06-1")))
may_rain <- may %>% group_by(gridnum, latitude, longitude)
totalmay <- may_rain%>% summarise(
  rain = sum(rainmm))
```

## 'summarise()' has grouped output by 'gridnum', 'latitude'. You can override  
## using the '.groups' argument.

```
may_sf <- st_as_sf(totalmay, coords = c("longitude", "latitude"), crs = 4326)

may_rain <- tm_shape(may_sf) +
  tm_dots(group = "rain", col = "rain", size = 1, palette = "-viridis", style="cont", breaks = c(0, 50, 100, 150, 200, 250)) +
  tm_layout( outer.margins = rep(0.06, 6), inner.margins = rep(0.08, 8), main.title = "Total Amount of Rainfall in April 2019")
may_rain
```

Total Amount of Rainfall(mm) in May (2019)



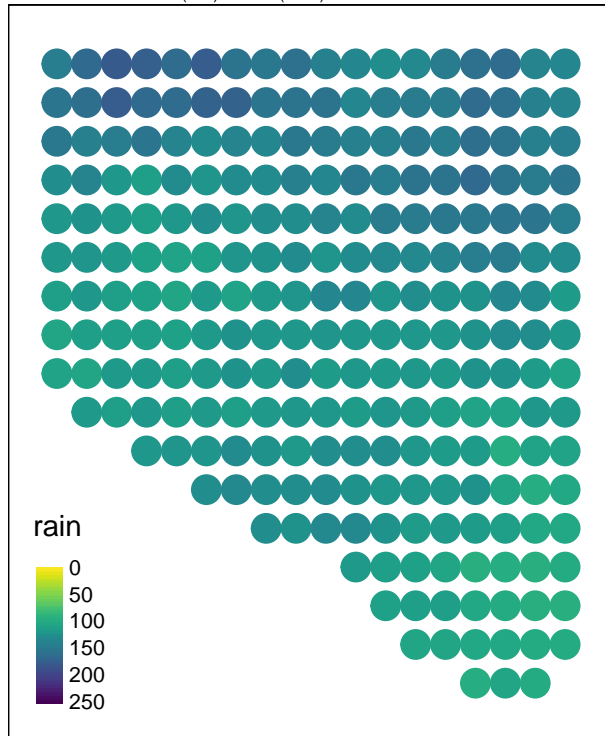
```
june <- filter(dayandpixel, between(Date, as.Date("2019-06-01"), as.Date("2019-07-1")))
june_rain <- may %>% group_by(gridnum, latitude, longitude)
totaljune <- june_rain%>% summarise(
  rain = sum(rainmm))
```

## 'summarise()' has grouped output by 'gridnum', 'latitude'. You can override  
## using the '.groups' argument.

```
june_sf <- st_as_sf(totaljune, coords = c("longitude", "latitude"), crs = 4326)

june_rain <- tm_shape(june_sf) +
  tm_dots(group = "rain", col = "rain", size = 1, palette = "-viridis", style="cont", breaks = c(0, 50, 100, 150, 200, 250)) +
  tm_layout(outer.margins = rep(0.06, 6), inner.margins = rep(0.08, 8), main.title = "Total Amount of Rainfall in June 2019")
june_rain
```

Total Amount of Rainfall(mm) in June (2019)



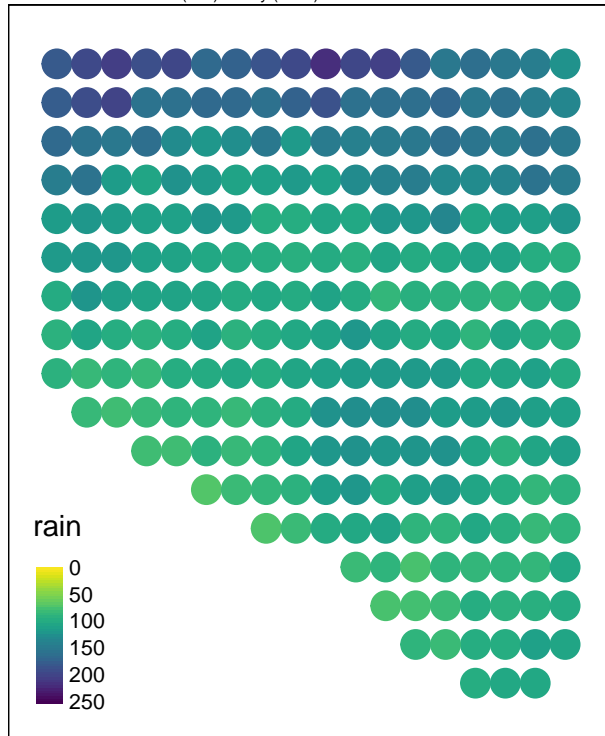
```
july <- filter(dayandpixel, between(Date, as.Date("2019-07-01"), as.Date("2019-08-1")))
july_rain <- july %>% group_by(gridnum, latitude, longitude)
totaljuly <- july_rain%>% summarise(
  rain = sum(rainmm))
```

## 'summarise()' has grouped output by 'gridnum', 'latitude'. You can override  
## using the '.groups' argument.

```
july_sf <- st_as_sf(totaljuly, coords = c("longitude", "latitude"), crs = 4326)
```

```
july_rain <- tm_shape(july_sf) +
  tm_dots(group = "rain", col = "rain", size = 1, palette = "-viridis", style = "cont", breaks = c(0, 50, 100, 150, 200, 250)) +
  tm_layout(outer.margins = rep(0.06, 6), inner.margins = rep(0.08, 8), main.title = "Total Amount of Rainfall in June 2019")
july_rain
```

Total Amount of Rainfall(mm) in July (2019)



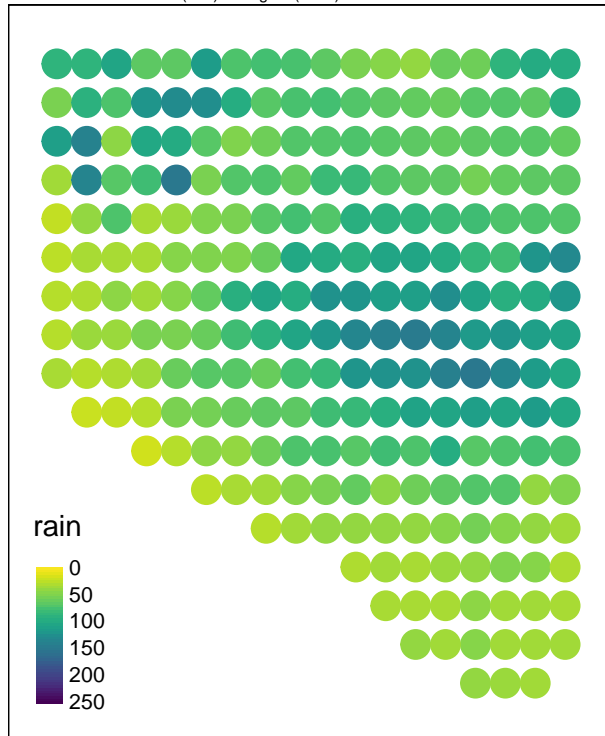
```
august <- filter(dayandpixel, between(Date, as.Date("2019-08-01"), as.Date("2019-09-1")))
aug_rain <- august %>% group_by(gridnum, latitude, longitude)
totalaug <- aug_rain%>% summarise(
  rain = sum(rainmm))
```

## 'summarise()' has grouped output by 'gridnum', 'latitude'. You can override  
## using the '.groups' argument.

```
aug_sf <- st_as_sf(totalaug, coords = c("longitude", "latitude"), crs = 4326)

aug_rain <- tm_shape(aug_sf) +
  tm_dots(group = "rain", col = "rain", size = 1, palette = "-viridis", style="cont", breaks = c(0, 50, 100, 150, 200, 250)) +
  tm_layout( outer.margins = rep(0.06, 6), inner.margins = rep(0.08, 8), main.title = "Total Amount of Rainfall in July 2019")
aug_rain
```

Total Amount of Rainfall(mm) in August (2019)



```
september <- filter(dayandpixel, between(Date, as.Date("2019-09-01"), as.Date("2019-10-1")))
sept_rain <- september %>% group_by(gridnum, latitude, longitude)
totalsept <- sept_rain%>% summarise(
  rain = sum(rainmm))
```

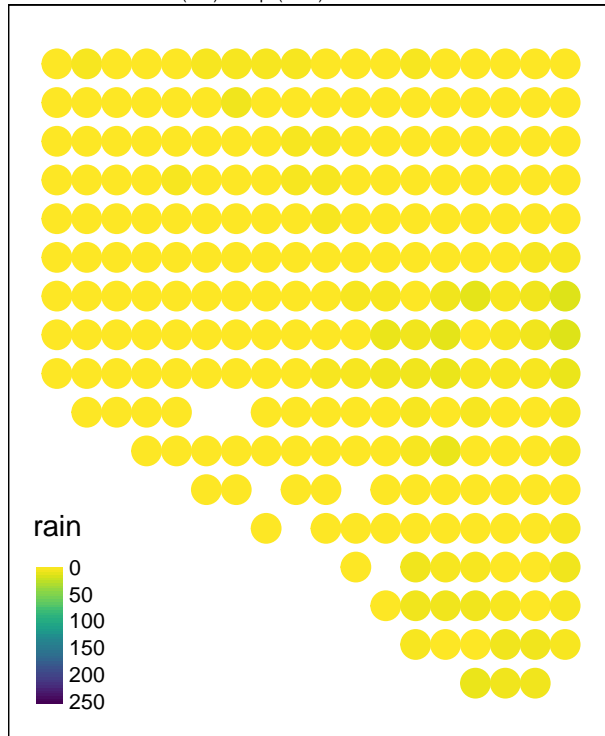
## 'summarise()' has grouped output by 'gridnum', 'latitude'. You can override  
## using the '.groups' argument.

```
sept_sf <- st_as_sf(totalsept, coords = c("longitude", "latitude"), crs = 4326)
```

```
sept_rain <- tm_shape(sept_sf) +
  tm_dots(group = "rain", col = "rain", size = 1, palette = "-viridis", style="cont", breaks = c(0, 50, 100, 150, 200, 250)) +
  tm_layout( outer.margins = rep(0.06, 6), inner.margins = rep(0.08, 8), main.title = "Total Amount of Rainfall in August 2019")
sept_rain
```



Total Amount of Rainfall(mm) in Sept (2019)



This is to show total rain (mm) over time from April to September 2019

```
#Animating the rain information per month
tmap_save(filename = "april.png", tm=april_rain,width=4,height=4,units="in",scale=1)
aprilgif <- image_read("april.png")

tmap_save(filename = "may.png", tm=may_rain,width=4,height=4,units="in",scale=1)
maygif <- image_read("may.png")

tmap_save(filename = "june.png", tm=june_rain,width=4,height=4,units="in",scale=1)
junegif <- image_read("june.png")

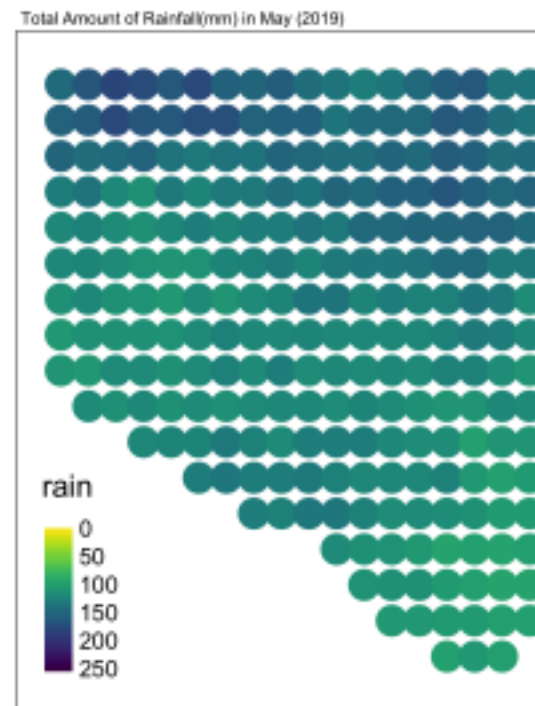
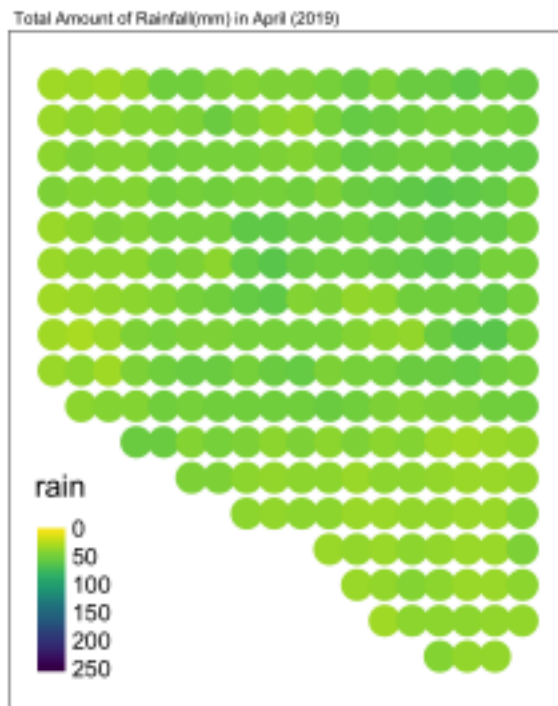
tmap_save(filename = "july.png", tm=july_rain,width=4,height=4,units="in",scale=1)
julygif <- image_read("july.png")

tmap_save(filename = "aug.png", tm=aug_rain,width=4,height=4,units="in",scale=1)
auggif <- image_read("aug.png")

tmap_save(filename = "sept.png", tm=sept_rain,width=4,height=4,units="in",scale=1)
septgif <- image_read("sept.png")

#this is putting each file together
img <- c(aprilgif, maygif, junegif, julygif, auggif, septgif)

image_append(image_scale(img, "x300"))
```



```
#this is actually creating the gif  
my.animation<-image_animate(image_scale(img, "400x400"), fps = 1, dispose = "previous")  
image_write(my.animation, "rainperpixel.gif")
```