Extracting Gridded Rainfall Data

Willem Vervoort, Chun Liang, Floris van Ogtrop

2018-06-05

Introduction

This document is related to the paper: Empirical study of land surface effects on local rainfall (Liang et al.) and describes the script to extract the gridded rainfall data from the Australian gridded rainfall data set from BOM. We have a paid licence for this data in the group and have the data stored on a shared server. The main contact for the data in the group is A/Prof Thomas Bishop. The formal metadata for the Gridded High Resolution Daily Rainfall is at the BOM website.

Defining the extent of the area and generating the points

The data is produced on a 0.05^o × 0.05^o grid. The study works with two areas, one in Queensland and one in NSW. The extent of these areas are defined below and the points generated to extract the rainfall data.

```
Lats_Qld <- seq(-28.075, -25.575, by = 0.05)
Longs_Qld <- seq(145.825, 148.625, by = 0.05)

Lats_NSW <- seq(-36.675, -35.075, by = 0.05)
Longs_NSW <- seq(147.075, 148.925, by = 0.05)

# use the vectors to generate a set of locations
Loc_NSW <- tibble(ID = numeric(length=length(Lats_NSW)*length(Longs_NSW)),
                  longs = numeric(length=length(Lats_NSW)*length(Longs_NSW)),
                  lats = numeric(length=length(Lats_NSW)*length(Longs_NSW)))
Loc_Qld <- tibble(ID = numeric(length=length(Lats_Qld)*length(Longs_Qld)),
                  longs = numeric(length=length(Lats_Qld)*length(Longs_Qld)),
                  lats = numeric(length=length(Lats_Qld)*length(Longs_Qld)))

for (i in 1:length(Lats_NSW)) {
  for (j in 1:length(Longs_NSW)) {
    Loc_NSW[(i-1)*length(Longs_NSW)+j,] <- c((i-1)*length(Longs_NSW)+j,
                                            Longs_NSW[j],Lats_NSW[i])
  }
}

for (i in 1:length(Lats_Qld)) {
  for (j in 1:length(Longs_Qld)) {
    Loc_Qld[(i-1)*length(Longs_Qld)+j,] <- c((i-1)*length(Longs_Qld)+j,
                                            Longs_Qld[j],Lats_Qld[i])
  }
}

# how many sites in each?
nrow(Loc_NSW)
```

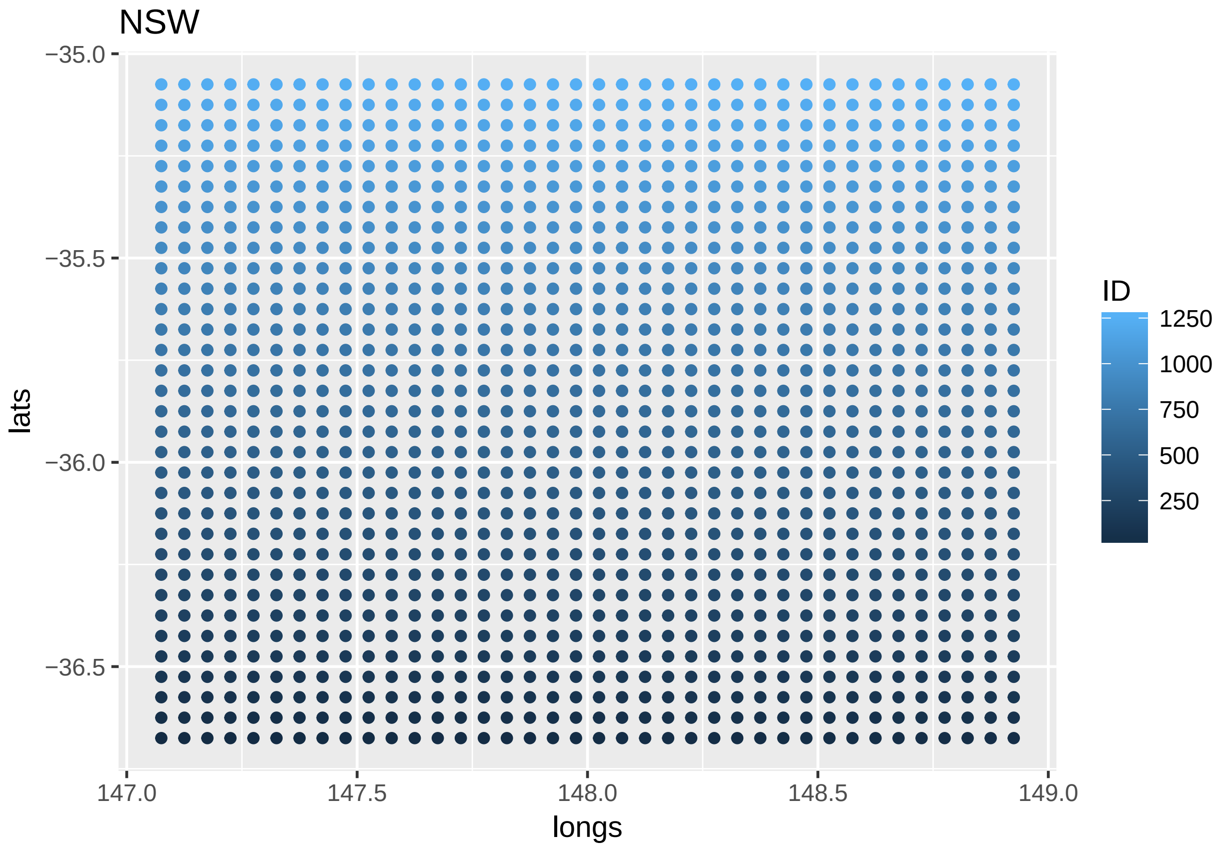
```
## [1] 1254
```

```
nrow(Loc_Qld)
```

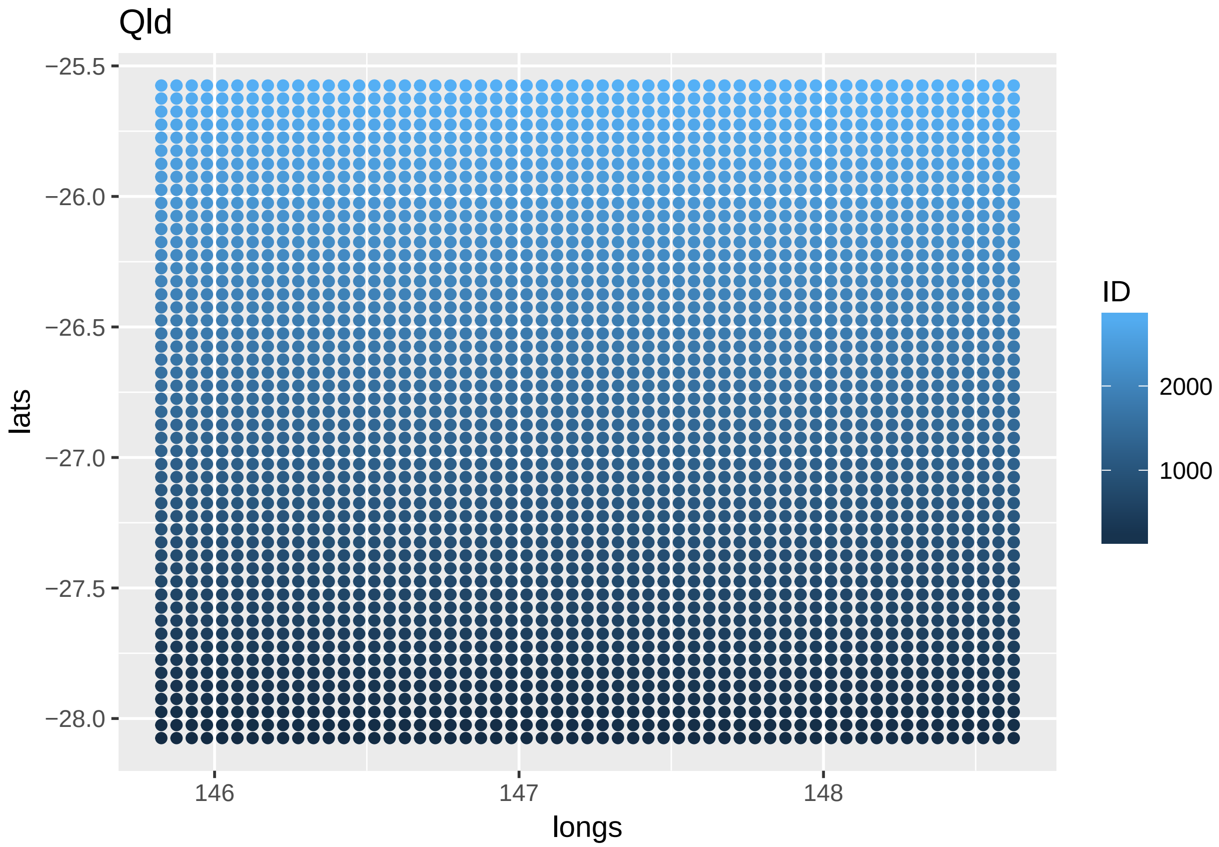
```
## [1] 2907
```

We can check whether this went OK by plotting the resulting lat and long values.

```
Loc_NSW %>%
  ggplot(aes(longs,lats, colour=ID)) + geom_point() + ggtitle("NSW")
```



```
Loc_Qld %>%
  ggplot(aes(longs,lats, colour=ID)) + geom_point() + ggtitle("Qld")
```



Now combine the two frames to make one final dataset for the extraction of the rainfall data

```
Loc <- rbind(Loc_NSW,Loc_Qld)
```

Extract rainfall data using Parallel

Extracting the rainfall takes time, so the easiest way to do this is using parallel computing, so we can use multiple cores on the PC.

```
# we need some packages

library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(raster)
```

```
## Loading required package: sp

##
## Attaching package: 'raster'

## The following object is masked from 'package:dplyr':
```

```
##
##     select

## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
library(rgdal)
```

```
## Warning: package 'rgdal' was built under R version 3.4.4

## rgdal: version: 1.2-20, (SVN revision 725)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.2.3, released 2017/11/20
##  Path to GDAL shared files: C:/Users/rver4657/Documents/R/win-library/3.4/rgdal/gdal
##  GDAL binary built with GEOS: TRUE
##  Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
##  Path to PROJ.4 shared files: C:/Users/rver4657/Documents/R/win-library/3.4/rgdal/proj
##  Linking to sp version: 1.2-7
```

```r
library(doParallel)
```

```
## Loading required package: foreach

##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when

## Loading required package: iterators

## Loading required package: parallel
```

```r
# directory where the data is stored
datadir <- "R:/grp-hgis/public/BOM_GriddedRainfallData"


# read in the list of decades
decades <- dir(paste(datadir,"Daily-rainfall",sep="/"), pattern="rainfall_")

# store to put data in
Store <- list()

# make the parallel cluster
cl <- makeCluster(4)
registerDoParallel(cl)
# run the foreach loop and export packages to slaves
Store<- foreach(i=1:(length(decades)-1),
                 .packages=c("raster","rgdal")) %dopar% {
  # read in the list of years
  Years <- dir(paste(datadir,"Daily-rainfall",decades[i],sep="/"))
  yearlist <- list()

  for (j in 1:length(Years)) {
    # read the grids
    files= list.files(paste(datadir,"Daily-rainfall",decades[i],Years[j],sep="/"),
                      pattern=".txt", full.names=TRUE)
    # stack the grids
    s <- stack(files)
    # extract the stations
    df <- extract(s,
                    SpatialPoints(as.data.frame(Loc[,2:3]),
  proj4string=CRS("+proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs")),
                df=TRUE, method='simple')
    # store in the list
    yearlist[[j]] <- df
  }
  yearlist
}
stopCluster(cl)
```

Once this is done, we need to do some combining and massaging to convert the list to a data frame that can be used in analyses.

```
output1 <- list()
for (i in 1:(length(decades)-1)) {
  output1[[i]] <- do.call(cbind,Store[[i]])
}

#remove ID column
for (i in 1:(length(decades)-1)) {
  remove <- grep("ID",names(output1[[i]]))
  output1[[i]] <- output1[[i]][,-as.numeric(remove)]
}

output2 <- do.call(cbind,output1)
# rename and save
Raindata <- output2
# save as an Rdata file
saveRDS(Raindata,file="Data/bom/rainfall/Raindata.RDS")
```