

# STAT585\_Lab2

Group 2: Zerui Zhang, Oscar Aguilar, Kellie McClermon

2/23/2019

```
library(sf)

## Linking to GEOS 3.6.1, GDAL 2.1.3, PROJ 4.9.3
library(ggspatial)

## Loading required package: ggplot2
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.2.1 --
## v tibble 1.4.2      v purrr 0.3.0
## v tidyr 0.8.2       v dplyr 0.7.8
## v readr 1.3.1      v stringr 1.3.1
## v tibble 1.4.2      v forcats 0.3.0

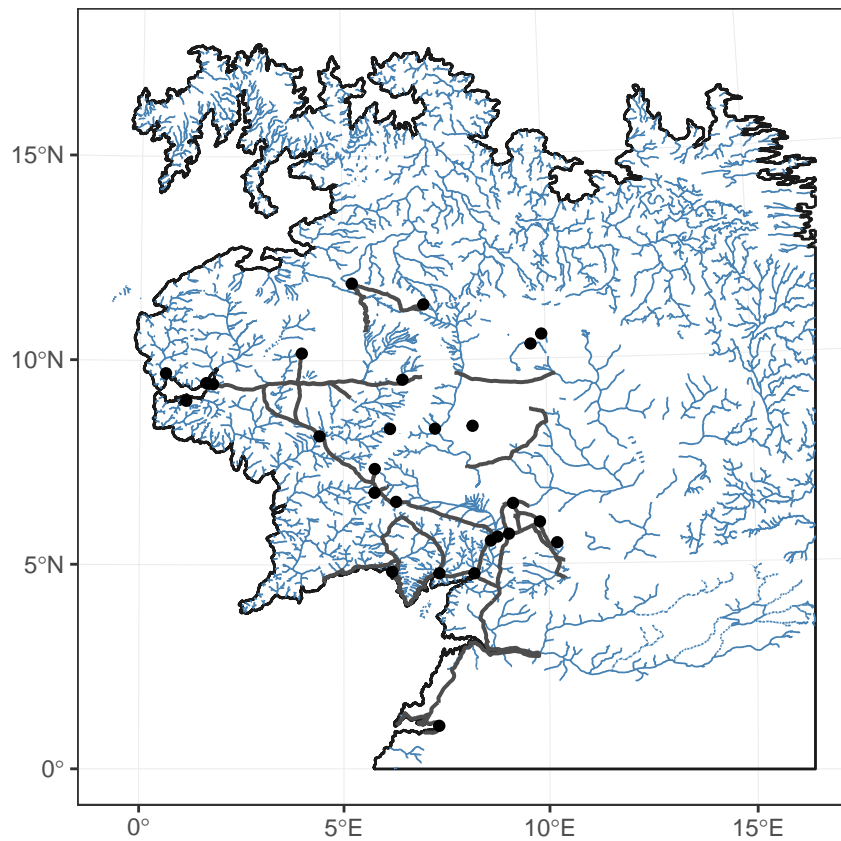
## Warning: package 'purrr' was built under R version 3.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()

library(ggplot2)
library(dplyr)

library(ggplot2)
library(sf)
library(tidyverse)
library(ggrepel)
library(ggspatial)

p <- ggplot() +
  geom_sf(data = read_sf("Data/ME-GIS-master/Coastline2.shp"),
    colour="grey10", fill="grey90") +
  geom_sf(data = read_sf("Data/ME-GIS-master/Rivers19.shp"),
    colour="steelblue", size=0.3) +
  geom_sf(data = read_sf("Data/ME-GIS-master/PrimaryRoads.shp"),
    size = 0.7, colour="grey30") +
  geom_sf(data = read_sf("Data/ME-GIS-master/Cities.shp")) +
  theme_bw()

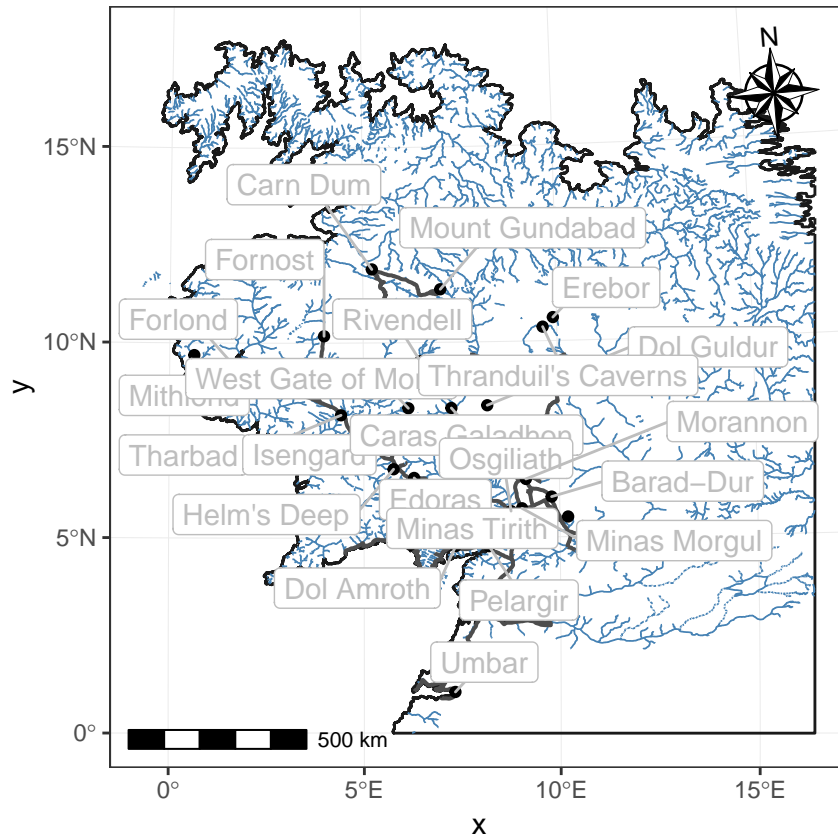
p
```



Add the labels

```
library(ggrepel)
p +
  geom_label_repel(data = read_sf("Data/ME-GIS-master/Cities.shp"),
    aes(label = Name, geometry = geometry),
    stat = "sf_coordinates",
    min.segment.length = 0,
    colour = "gray",
    segment.colour = "gray")+
  annotation_scale() +
  annotation_north_arrow(which_north = "true", location = "true", style = north_arrow_nautical())
```

## Warning: Removed 4 rows containing missing values (geom\_label\_repel).



Australia shapefile

```
ozbig <- read_sf("Data/gadm36_AUS_shp/gadm36_AUS_1.shp")
oz_st <- maptools::thinnedSpatialPoly(
  as(ozbig, "Spatial"), tolerance = 0.1,
  minarea = 0.001, topologyPreserve = TRUE)
oz <- st_as_sf(oz_st)
```

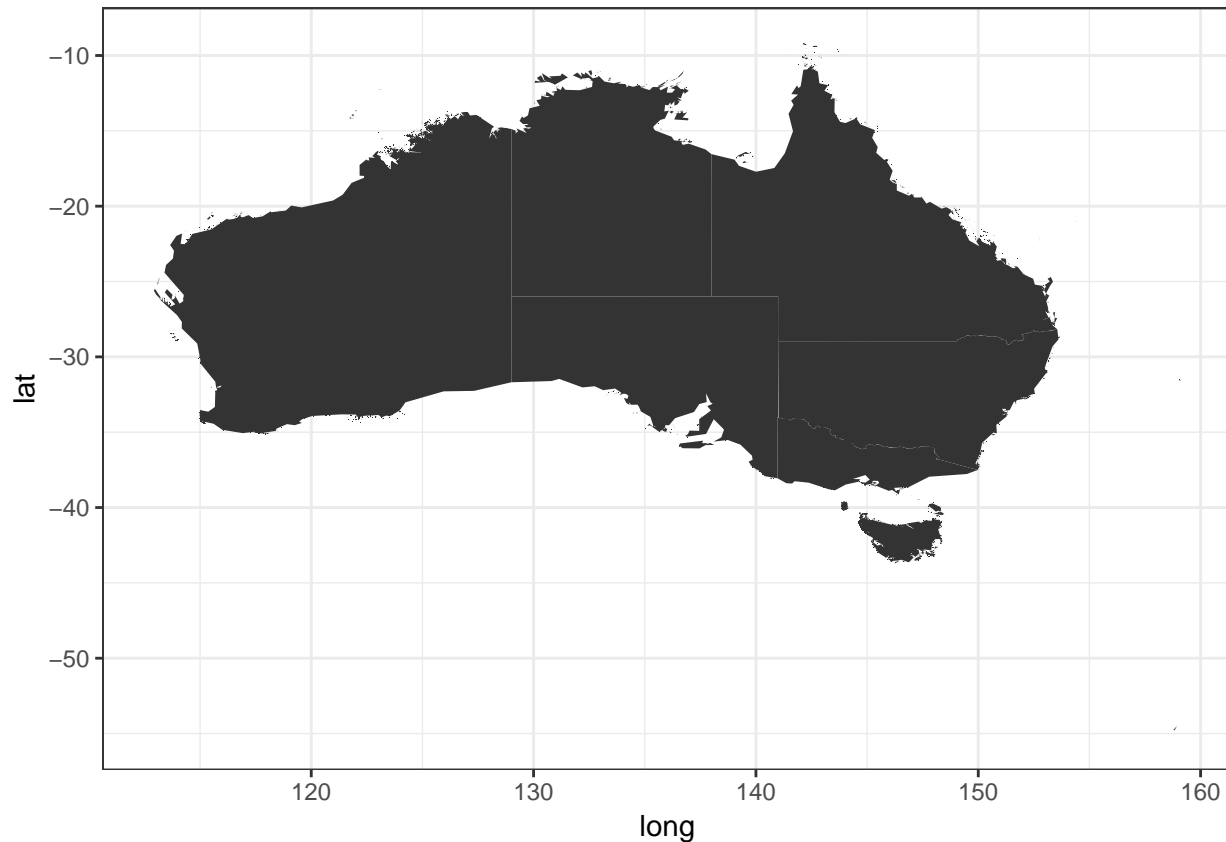
```
## Help calculate the group.
## Input: oz$geometry
helper.group <- function(geo){
  geo %>%
    unlist(., recursive = FALSE) %>%
    unlist(., recursive = FALSE) -> dd
  countgroupprep <- unlist(map(dd, nrow))
  num_group <- length(countgroupprep)
  rep(1:num_group, time = countgroupprep)
}
```

```
## Help calculate the order.
## Input: oz$geometry[[i]]
helper.order <- function(geol){
  geol %>%
    unlist(., recursive = FALSE) -> d
  longlat <- do.call(rbind, d)
  order_num <- sum(unlist(map(d, nrow)))
  order <- seq(1:order_num)
  cbind(longlat, order)
```

```

}
res <- map(oz$geometry, .f=helper.order)
ress <- do.call(rbind, res)
group <- helper.group(oz$geometry)
ress <- cbind(ress, group)
colnames(ress) <- c("long", "lat", "order", "group")
ress <- as.data.frame(ress)
ress %>% ggplot(aes(x=long,y=lat,group=group)) + geom_polygon() + theme_bw()

```



We pick France

```

frabig <- read_sf("Data/gadm36_FRA_shp/gadm36_FRA_1.shp")
fra_st <- maptools::thinnedSpatialPoly(
  as(frabig, "Spatial"), tolerance = 0.1,
  minarea = 0.001, topologyPreserve = TRUE)
fr <- st_as_sf(fra_st)
frres <- map(fr$geometry, .f=helper.order)
frress <- do.call(rbind, frres)
frgroup <- helper.group(fr$geometry)
frress <- cbind(frress, frgroup)
colnames(frress) <- c("long", "lat", "order", "group")
frress <- as.data.frame(frress)
frress %>% ggplot(aes(x=long,y=lat,group=group)) + geom_polygon() + theme_bw()

```

