HurricanMaps

Bruce Mallory

11/2/2020

(1) Read in all the data

```
addRepo("geanders")
data("hurr_tracks")
data("rain")
county.longlat <- map_data("county")
data(county.fips)

floyd_track <- hurr_tracks %>% filter(storm_id=="Floyd-1999") %>%
  filter(longitude <= -65) %>% filter(latitude >= 23)
floyd_rain <- rain %>% filter(storm_id=="Floyd-1999")

allison_track <- hurr_tracks %>% filter(storm_id=="Allison-2001")
allison_rain <- rain %>% filter(storm_id=="Allison-2001")

#for ggplots
mp_states <- c("texas", "oklahoma", "kansas",
               "iowa", "missouri", "arkansas", "louisiana",
               "alabama", "mississippi", "georgia", "florida",
               "tennessee","kentucky", "indiana",
               "wisconsin", "michigan", "illinois",
               "ohio", "west virginia", "pennsylvania",
               "south carolina", "north carolina",
               "virginia", "delaware", "maryland",
               "new jersey", "district of columbia",
               "new york", "connecticut", "rhode island",
               "massachusetts", "vermont", "new hampshire",
               "maine")
states_tm <- c("TX", "OK", "KS", "IA", "MO", "AR", "LA", "AL", "MS", "GA", "FL",
               "TN", "KY", "IN", "WI", "MI", "IL", "OH", "WV", "PA", "SC", "NC",
               "VA", "DE", "MD", "NJ", "DC", "NY", "CT", "RI", "MA", "VT", "NH",
               "ME")
#for tmaps
states <- map_data('state', region = mp_states)
```

(2) Calculate total rain/fip, then make rain variable categorical (9 levels for FLoyd, 2 levels for Allison)

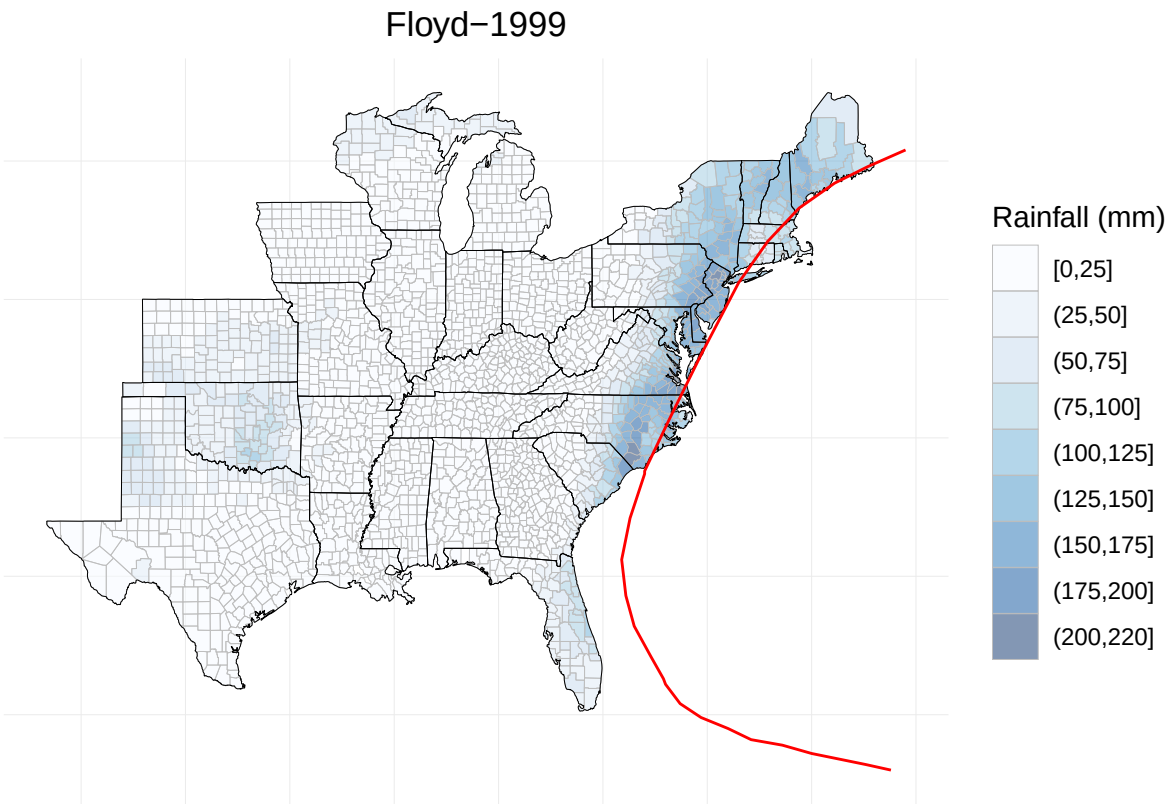
```
floyd_rain %<>% group_by(fips) %>% summarise(rain = sum(precip), .groups="drop")
floyd_rain %<>% as.data.frame()
for (i in 1:dim(floyd_rain)[1]){
  floyd_rain$rain[i] <- floyd_rain$rain[i]%/%25
}
floyd_rain$rain <- ordered(floyd_rain$rain,labels = c("[0,25]","(25,50]","(50,75]",
                                                      "(75,100]","(100,125]","(125,150]",
                                                      "(150,175]","(175,200]","(200,220]"))

allison_rain %<>% group_by(fips) %>% summarise(rain = sum(precip), .groups="drop")
allison_rain %<>% as.data.frame()
for (i in 1:dim(allison_rain)[1]){
  if (allison_rain$rain[i] < 175){
    allison_rain$rain[i] <- 0
  }
  else {allison_rain$rain[i] <- 1}
}
allison_rain$rain <- ordered(allison_rain$rain, labels = c("Unexposed","Exposed"))
```

(3) Join files to get a single file with long, lat, rain (which will be used to plot the data)

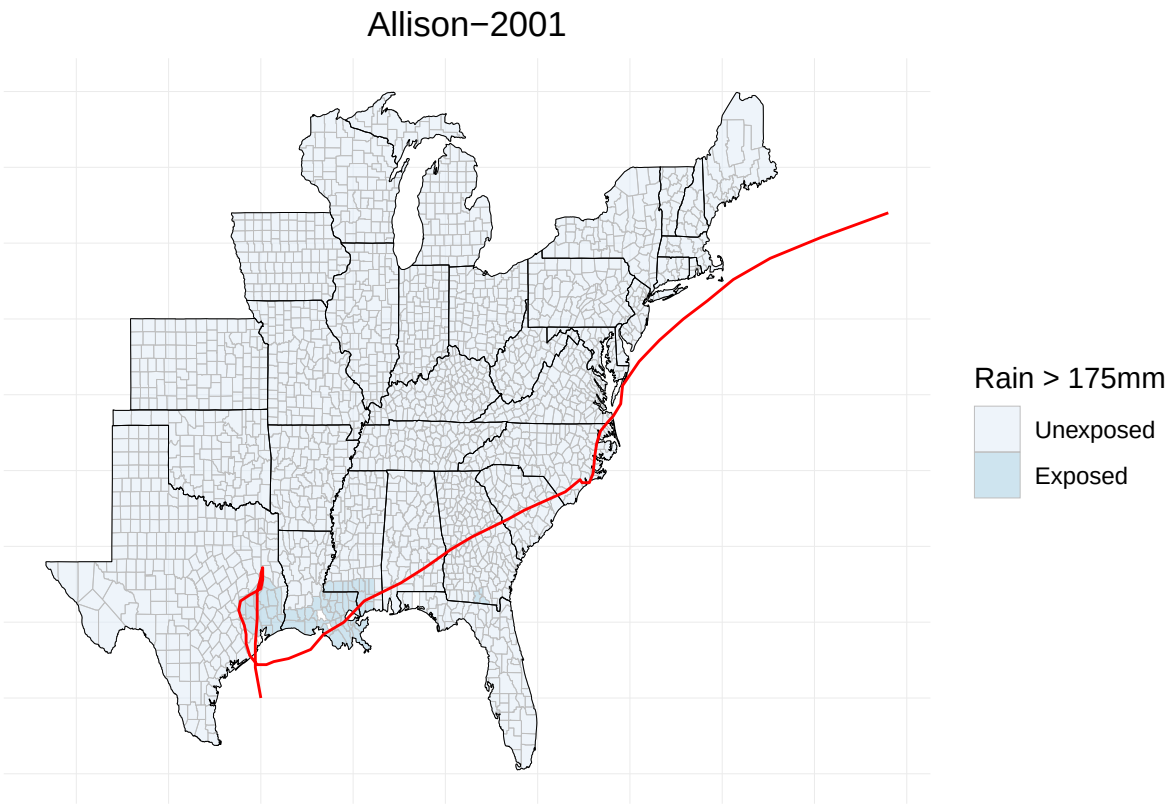
```
#(1) split "polyname" into "region" and "subregion" so: county.fips and county.longlat can be joined
county.fips %<>%
  separate(polyname, c("region", "subregion"), "," )
county <- left_join(county.longlat,county.fips, by = c("region", "subregion"))
#(2) make "fips" numeric so: "county" and "xxx_rain" can be joined
floyd_rain$fips <- as.numeric(floyd_rain$fips)
floyd_rain %<>% na.omit()
floyd_rain1 <- left_join(floyd_rain, county, by = "fips")
allison_rain$fips <- as.numeric(allison_rain$fips)
allison_rain <- left_join(allison_rain, county, by = "fips")
```

(4) Use ggplot to plot first the rain, then the states, then the track (and format the graph)

```
ggplot() +
  geom_polygon(data=floyd_rain1, aes(x=long, y=lat, group=group, fill=rain), color="grey", size=0.05, al
  scale_fill_brewer(name="Rainfall (mm)", palette="Blues") +
  geom_polygon(data=states, aes(x=long, y=lat, group=group), fill="transparent", color="black", size=0.
  labs(title="Floyd-1999") +
  theme_minimal() +
  labs(x="", y="") +
  theme(plot.title = element_text(hjust = 0.5), element_line(size = 0), axis.text =element_blank()) +
  geom_path(data=floyd_track, aes(x=longitude, y=latitude), color="red", size=0.5)
```



```
ggplot() +
  geom_polygon(data=allison_rain, aes(x=long, y=lat, group=group, fill=rain), color="grey", size=0.05, a
  scale_fill_brewer(name="Rain > 175mm", palette="Blues") +
  geom_polygon(data=states, aes(x=long, y=lat, group=group), fill="transparent", color="black", size=0.
  ggtitle("Allison-2001")  +
  theme_minimal() +
  labs(x="", y="") +
  theme(plot.title = element_text(hjust = 0.5), element_line(size = 0), axis.text =element_blank()) +
  geom_path(data=allison_track, aes(x=longitude, y=latitude), color="red", size=0.5)
```



(5) AND here's my attmept to do the same graphs using tmap

```
## Transfer data into spatial version
states2 <- states(class="sf")
```

```
##   |                                                                      |
```

```
county2 <- counties(class="sf")
```

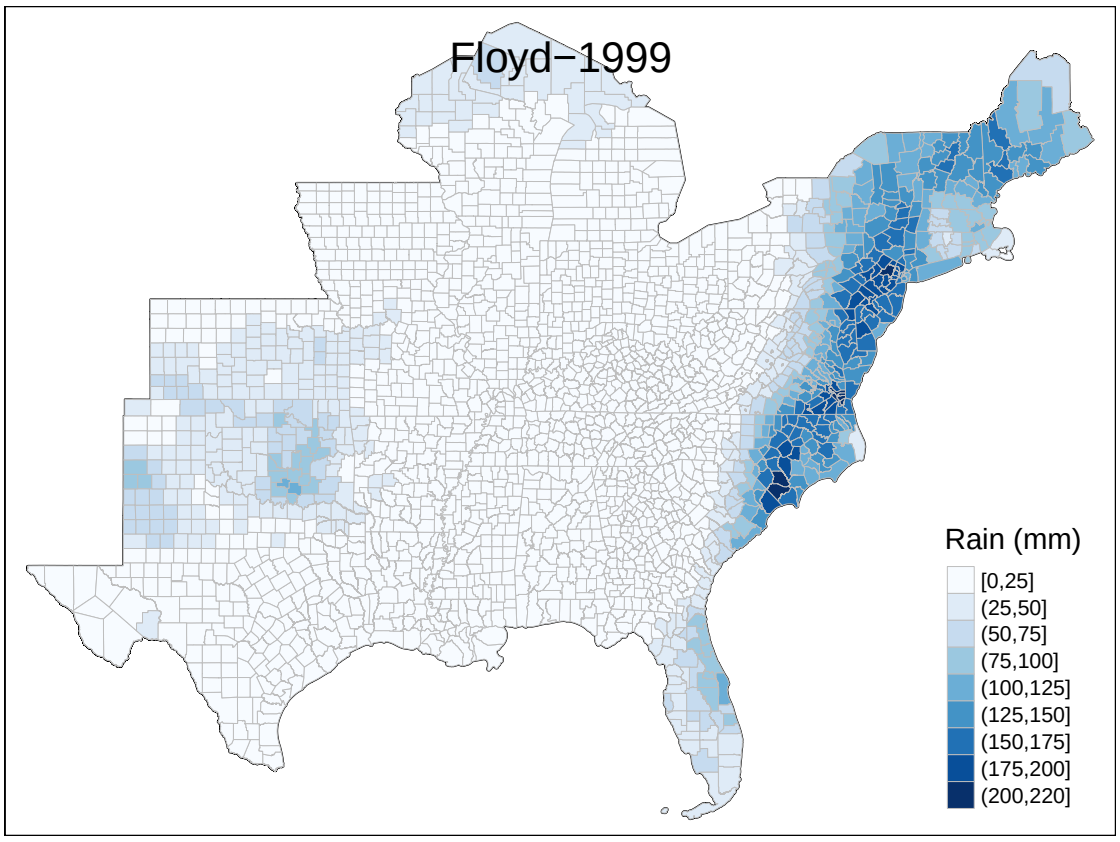
```
##   |                                                                      |
```

```
states2 %<>% filter(STUSPS %in% states_tm)
county2 %<>% filter(STATEFP %in% states2$STATEFP)

county2$fips <-as.numeric(county2$GEOID)
floyd_rain2 <- left_join(floyd_rain, county2)
```

```
## Joining, by = "fips"
```

```
floyd_rain2 <- st_as_sf(floyd_rain2)

tm_shape(states2) +
  tm_borders(col="black", lwd=1) +
  tm_layout(title="Floyd-1999", title.position = c("center", "top")) +
tm_shape(floyd_rain2) +
  tm_borders(col="grey",lwd=0.1) +
  tm_fill("rain", id="fips", title = "Rain (mm)", palette = "Blues")
```

```
## Warning: The shape floyd_rain2 contains empty units.
```


Floyd−1999
Rain (mm)
[0,25]
(25,50]
(50,75]
(75,100]
(100,125]
(125,150]
(150,175]
(175,200]
(200,220]

NOTES:

tmaps ended up being REALLY slow. And I had to restart my machine a couple of times, after RStudio froze. Which leads me to believe that there is something wrong with the shape files that I am feeding tmap. (BUT, the "simple" plot of the state outlines took a significant amount of time to render.) I did not attempt to get the strom track for Floyd on the map, and I did not attempt Allison.