STAT 510: Spatiotemporal Stats

Spatiotemporal Stats Lab 1

Prof. Taylor-Rodriguez

```
knitr::opts_chunk$set(echo = TRUE,cache = TRUE)
library(STRbook)
library(ggplot2)
library(dplyr)
library(tidyr)
library(tidyverse)
library(sp)
library(spacetime)
library(lubridate)
```

Let's get the Lab started

We'll be working with the following datasets

1. *Station.dat*: (328×3) station info
2. *Times_1990.dat*: (1461×4) daily time stamps between Jan 1st 1990 and Dec 30 1993
3. *Tmax_1990.dat*: (1461×328) Daily Tmax by station (NA's$\equiv -9999$)
4. *Tmin_1990.dat*: (1461×328) Daily Tmin by station (NA's$\equiv -9999$)
5. *Precip_1990.dat*: (1461×328) Daily Precip by station (NA's$\equiv -9999.90001$)
6. *Precip_1990.dat*: (1461×328) Daily Precip by station (NA's$\equiv -9999.989998$)

Data Wrangling

Data Wrangling and Visualization

First, let's load into R, combine and rearrange into long format these datasets.

This can help summarize and visualize the data

Loading data tables into R

```r
#locations
locs <- read.table(system.file("extdata","Stationinfo.dat",
                               package = "STRbook"),
                   col.names = c("id", "lat", "lon"))

#times
Times <- read.table(system.file("extdata","Times_1990.dat",
                                package = "STRbook"),
                    col.names = c("julian", "year", "month", "day"))

#Actual data files
Tmax <- read.table(system.file("extdata", "Tmax_1990.dat",
                               package = "STRbook"))
Tmin <- read.table(system.file("extdata", "Tmin_1990.dat",
                               package = "STRbook"))
Precip <- read.table(system.file("extdata", "Precip_1990.dat",
                               package = "STRbook"))

#name columns (station id's) in data tables
names(Tmax) <- names(Tmin) <- names(Precip) <- locs$id
```

Convert data into long format

```r
#using dplyr functions:
Tmax_long <- bind_cols(Times,Tmax) %>% #add time stamps to Tmax
  gather(key="id", value="values",
         -julian, -year, -month, -day) %>% #make into long format
  mutate(proc="Tmax") #add variable label

head(Tmax_long,4)
```

```
##   julian year month day   id values proc
## 1 726834 1990     1   1 3804     35 Tmax
## 2 726835 1990     1   2 3804     42 Tmax
## 3 726836 1990     1   3 3804     49 Tmax
## 4 726837 1990     1   4 3804     59 Tmax
```

Note: the `gather` function has been replaced by `pivot_long`... but for now we'll stick to what they use in the book

Convert data into long format

```r
#rinse and repeat with Tmin and Precip
Tmin_long <- bind_cols(Times,Tmin) %>%
  gather(key="id", value="values", -julian, -year, -month, -day) %>%
  mutate(proc="Tmin")

Precip_long <- bind_cols(Times,Precip) %>%
  gather(key="id", value="values", -julian, -year,
         -month, -day)  %>%
  mutate(proc="Precip")
```

Combine data and remove NA's

```r
#combine all data into single
NOAA_df_1990 <- bind_rows(Tmax_long,
                          Tmin_long,
                          Precip_long)
#remove NA's
NOAA_df_1990 <- NOAA_df_1990 %>%
  filter(values>-9999 & (proc%in%c("Tmax","Tmin")) |
           values>-99 & proc=="Precip")
#remove previous datasets
rm(list=c("Tmax_long","Tmin_long","Precip_long"))
```

Some simple data summaries

```
#overall means by variable ("proc" in data)
summ <- NOAA_df_1990 %>%
  group_by(proc) %>%
  summarise(proc_means=mean(values))
head(summ)
```

```
## # A tibble: 3 x 2
##   proc   proc_means
##   <chr>       <dbl>
## 1 Precip      0.117
## 2 Tmax       65.5
## 3 Tmin       45.4
```

Some simple data summaries

```
#yearly means by variable
summ <- NOAA_df_1990 %>%
  group_by(proc,year) %>%
  summarise(proc_means=mean(values))
head(summ,5)
```

```
## # A tibble: 5 x 3
## # Groups:   proc [2]
##   proc    year proc_means
##   <chr>  <int>      <dbl>
## 1 Precip  1990      0.126
## 2 Precip  1991      0.116
## 3 Precip  1992      0.111
## 4 Precip  1993      0.116
## 5 Tmax    1990     67.4
```

More involved summaries

Mean number of dry days in June each year by station

```
summ <- NOAA_df_1990 %>%
  filter(proc=="Precip",month==6,values==0) %>%
  group_by(id,year,month) %>%
  summarise(numdry=n())
head(summ)
```

```
## # A tibble: 6 x 4
## # Groups:   id, year [6]
##   id     year month numdry
##   <chr> <int> <int>  <int>
## 1 13865  1990     6     23
## 2 13865  1991     6     19
## 3 13865  1992     6     20
## 4 13865  1993     6     23
## 5 13866  1990     6     19
## 6 13866  1991     6     22
```

More involved summaries

Mean number of dry days in June each year

```
summ <- summ %>%
  ungroup() %>%
  group_by(year,month) %>%
  summarise(mu_noprecip=mean(numdry)) %>%
  select(-month)
head(summ)
```

```
## # A tibble: 4 x 2
## # Groups:   year [4]
##    year mu_noprecip
##   <int>       <dbl>
## 1  1990        20.5
## 2  1991        22.2
## 3  1992        19.7
## 4  1993        18.7
```

Combining datasets using *keys*

Let's add the station location information to the combined data

```
locs <- locs %>% mutate(id=as.character(id))
NOAA_df_1990 <- NOAA_df_1990 %>%
  left_join(locs,by="id")

head(NOAA_df_1990,5)
```

```
##   julian year month day   id values proc   lat       lon
## 1 726834 1990     1   1 3804     35 Tmax 39.35 -81.43333
## 2 726835 1990     1   2 3804     42 Tmax 39.35 -81.43333
## 3 726836 1990     1   3 3804     49 Tmax 39.35 -81.43333
## 4 726837 1990     1   4 3804     59 Tmax 39.35 -81.43333
## 5 726838 1990     1   5 3804     41 Tmax 39.35 -81.43333
```

Note: `left_join` keeps all observations in left dataset and finds matches according to the *key* (here *"id"*) in the dataset on the right hand side. Other options available.

Create *date* variable from *year*, *month*, *day*

Let's add the station location information to the combined data

```
NOAA_df_1990 <- NOAA_df_1990 %>%
  unite("date",year,month,day,remove=F) %>%
  mutate(date=ymd(date))

head(NOAA_df_1990,5)
```

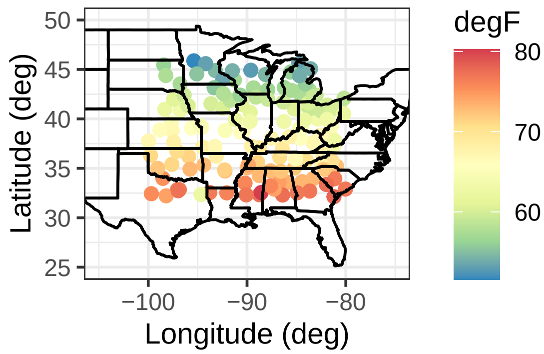
```
##   julian       date year month day   id values proc   lat       lon
## 1 726834 1990-01-01 1990     1   1 3804     35 Tmax 39.35 -81.43333
## 2 726835 1990-01-02 1990     1   2 3804     42 Tmax 39.35 -81.43333
## 3 726836 1990-01-03 1990     1   3 3804     49 Tmax 39.35 -81.43333
## 4 726837 1990-01-04 1990     1   4 3804     59 Tmax 39.35 -81.43333
## 5 726838 1990-01-05 1990     1   5 3804     41 Tmax 39.35 -81.43333
```

Visualization

Empirical Spatial Means Plot

For geostatistical data

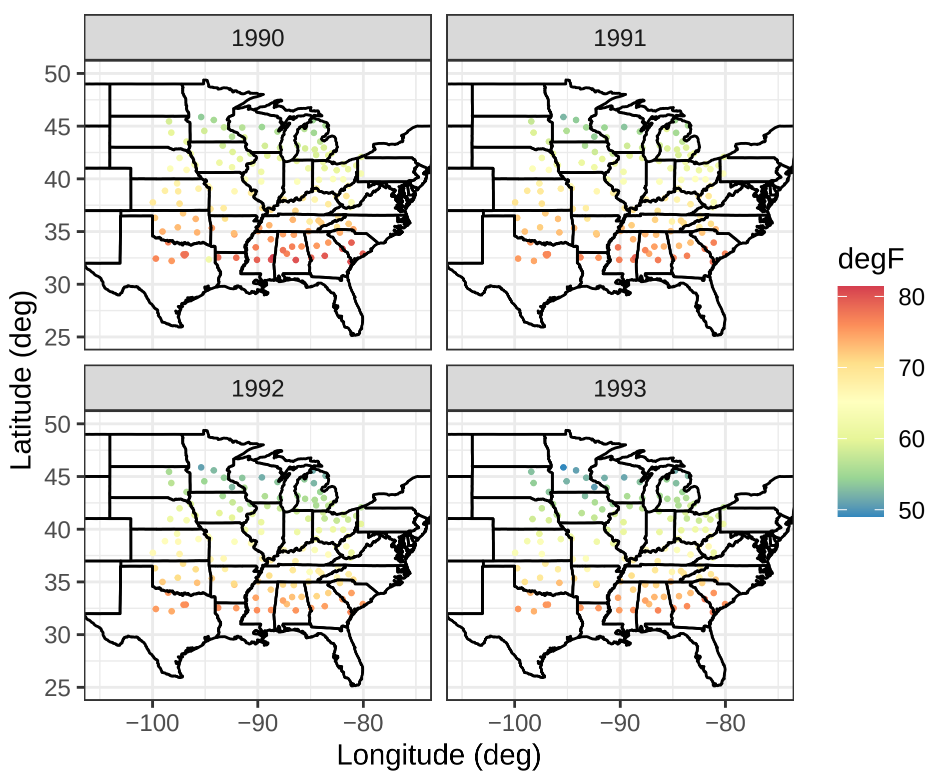
```
NOAA_df_1990 %>%
  filter(proc=="Tmax") %>%
  group_by(id,lon,lat) %>%
  summarise(z=mean(values)) %>%
  ggplot() +
  geom_point(aes(x=lon,y=lat,colour=z),size=2) +
  col_scale(name = "degF") +
  xlab("Longitude (deg)") +
  ylab("Latitude (deg)") +
  geom_path(data = map_data("state"),# state boundaries
            aes(x = long, y = lat, group = group)) +
  coord_fixed(xlim = c(-105, -75),
              ylim= c(25,50)) +
  theme_bw()
```



Empirical Spatial Means Plot (By Year)

For geostatistical data

```
NOAA_df_1990 %>%
  filter(proc=="Tmax") %>%
  group_by(id,lon,lat,year) %>%
  summarise(z=mean(values)) %>%
  ggplot() +
  geom_point(aes(x=lon,y=lat,colour=z),size=0.5) +
  col_scale(name = "degF") +
  xlab("Longitude (deg)") +
  ylab("Latitude (deg)") +
  geom_path(data = map_data("state"),# state boundaries
            aes(x = long, y = lat, group = group)) +
#  facet_grid(.~year) + #,rows=2,cols=2
  facet_wrap(.~year,nrow=2,ncol=2) + #
  coord_fixed(xlim = c(-105, -75),
              ylim= c(25,50)) +
  theme_bw()
```



Empirical Spatial Covariances and Plots

For geostatistical data

Let's work with the Tmax data between May and September of 1993

```
Tmax_long <- NOAA_df_1990 %>% # now subset the data
  filter(proc == "Tmax" &      # only max temperature
           month %in% 5:9 &    # May to September
           year == 1993) %>%   # year 1993
  mutate(t=as.integer(julian-min(julian-1))) #create time variable
```

Empirical Spatial Covariances and Plots

But first, de-trend the data

```
lm1 <- lm(values ~ lat + t + I(t^2), # fit a linear model
          data = Tmax_long)
Tmax_long$residuals <- residuals(lm1) # store the residuals
```

Extract the spatial locations

```
spat_df <- filter(Tmax_long, t == 1) %>% # lon/lat coords of stations
  select(lon, lat) %>% # select lon/lat only
  arrange(lon, lat) # sort ascending by lon/lat
m <- length(unique(Tmax_long$id)) # number of stations
```

Empirical Spatial Covariances and Plots

Put data in *space-wide* format

```
X <- Tmax_long %>% select(lon, lat, residuals, t) %>% # select columns
spread(t, residuals) %>% select(-lon, -lat) %>% t()
```

Calculate lag-0 and lag-1 covariance matrices (use `complete.obs`)

```
Lag0_cov <- cov(X, use = 'complete.obs')
Lag1_cov <- cov(X[-1, ], X[-nrow(X),],use = 'complete.obs')
```

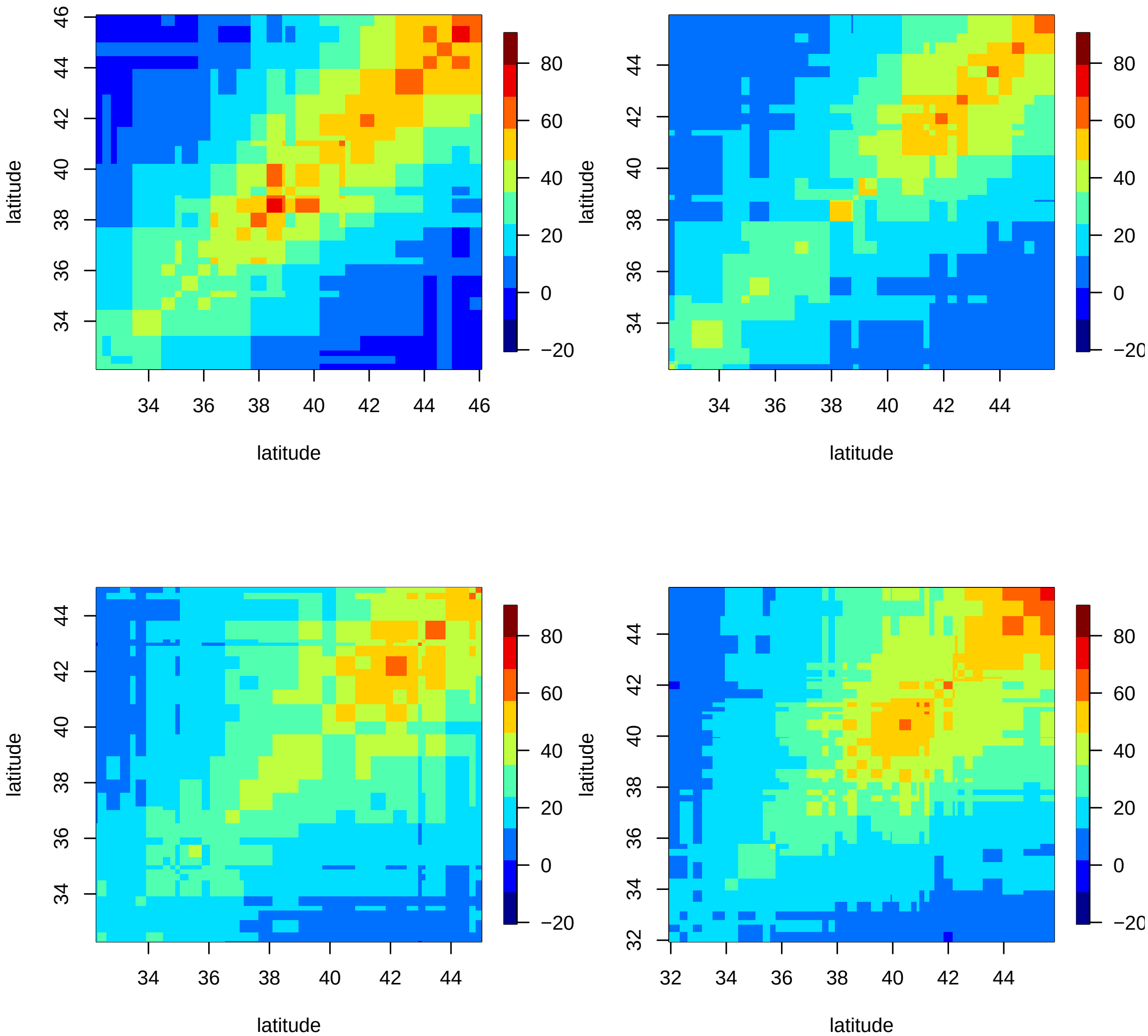
Empirical Spatial Covariances and Plots

```
# assign an index to each station
spat_df$n <- 1:nrow(spat_df)

# range of lon coordinates
lim_lon <- range(spat_df$lon)

# create 4 long. strip boundaries
lon_strips <- seq(lim_lon[1],lim_lon[2],length = 5)
spat_df$lon_strip <- cut(spat_df$lon, # bin the lon into
                         lon_strips, # their respective bins
                         labels = FALSE, # don't assign labels
                         include.lowest = TRUE) # include edges
```

Empirical Spatial Means Plot (By Year)

Lag 0 covariances

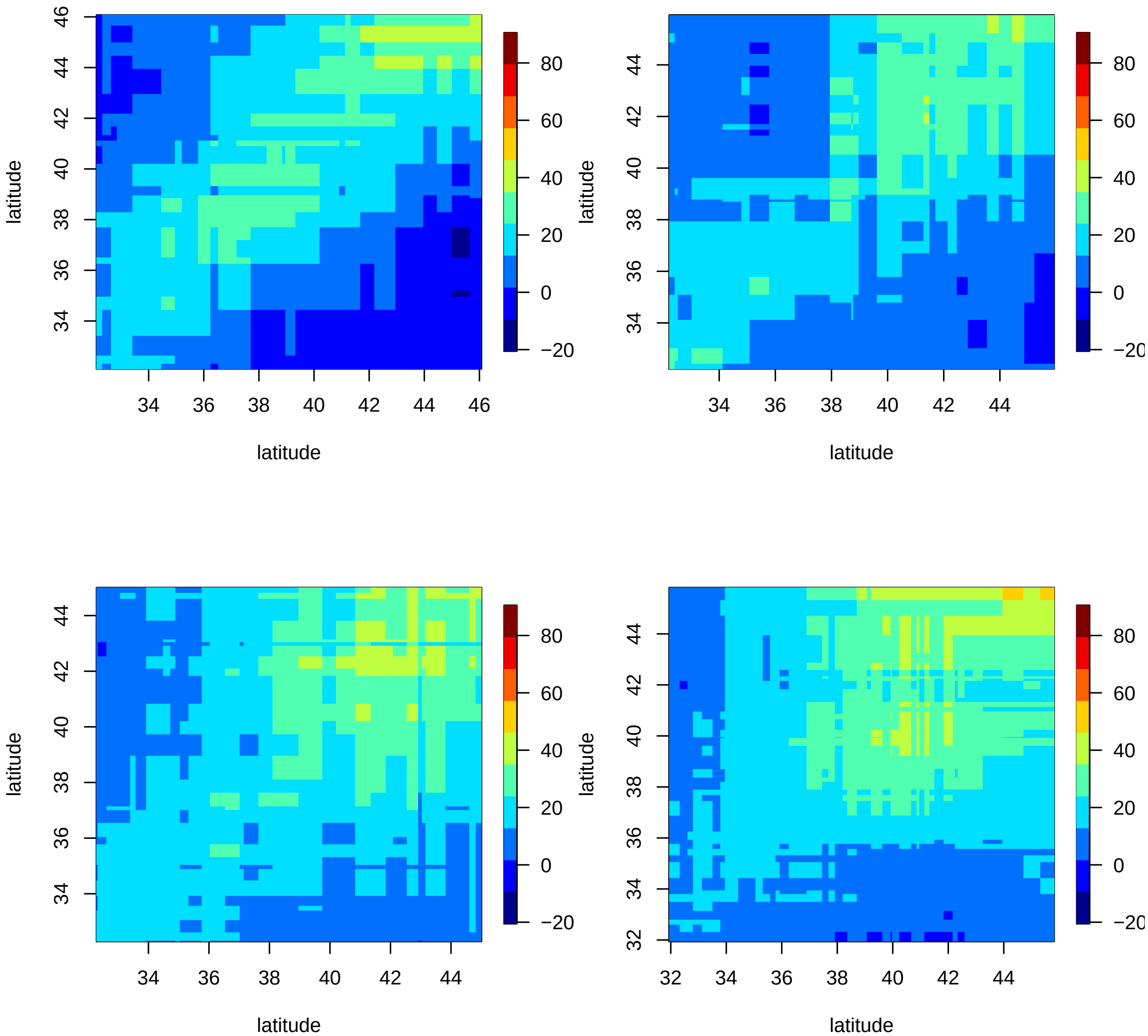
```
par(mfrow=c(2,2))
plot_cov_strips(Lag0_cov, spat_df) # plot the lag-0 matrices
```



Empirical Spatial Means Plot (By Year)

Lag 1 covariances

```
par(mfrow=c(2,2))
plot_cov_strips(Lag1_cov, spat_df) # plot the lag-1 matrices
```



Empirical Spatial Means Plot

For areal data

```
data("BEA", package = "STRbook") #income data
head(BEA %>% select(-Description), 3)
```

```
##          NAME10 X1970 X1980 X1990
## 6     Adair, MO  2723  7399 12755
## 9    Andrew, MO  3577  7937 15059
## 12 Atchison, MO  3770  5743 14748
```

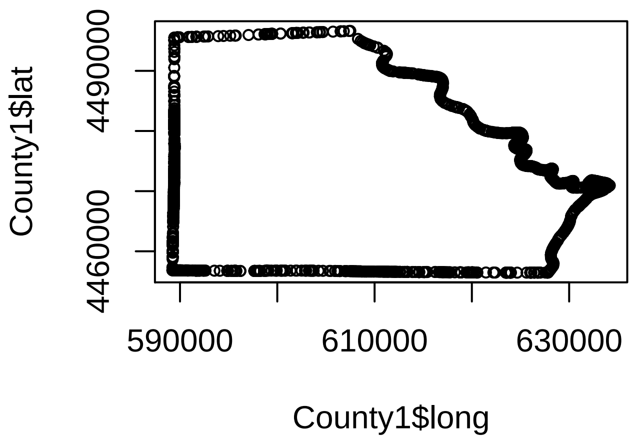
```
data("MOcounties", package = "STRbook")  #county geo data
```

Empirical Spatial Means Plot

For areal data

Join MO county boundary data with income

```
MOcounties <- left_join(MOcounties, BEA, by = "NAME10")

County1 <- filter(MOcounties, NAME10 == "Clark, MO")
plot(County1$long, County1$lat, cex=0.7)
```



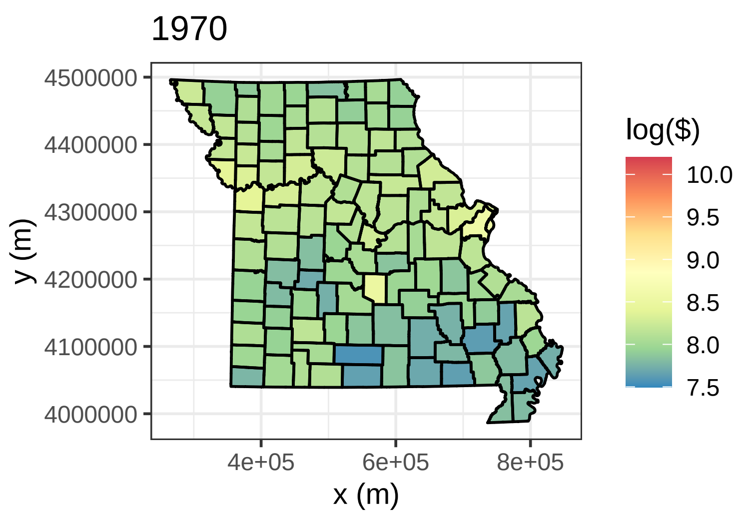
Empirical Spatial Means Plot

For areal data

```
ggplot(MOcounties) +
  geom_polygon(aes(x = long, y = lat, # county boundary
                   group = NAME10, # county group
                   fill = log(X1970))) + # log of income
  geom_path(aes(x = long, y = lat, # county boundary
                group = NAME10)) + # county group
  fill_scale(limits = c(7.5,10.2),name = "log($)") +
  coord_fixed() + ggtitle("1970") + # annotations
  xlab("x (m)") + ylab("y (m)") + theme_bw()
```

Empirical Spatial Means Plot

For areal data



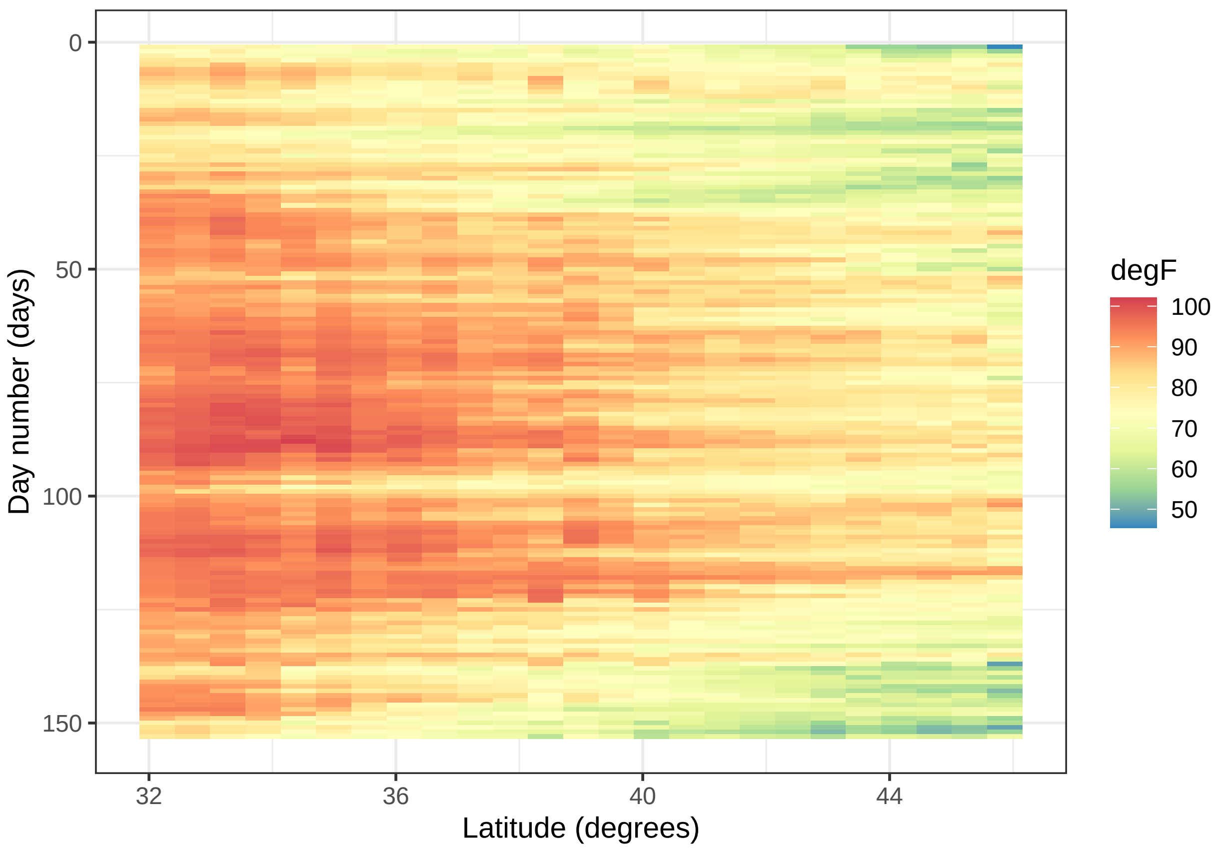
Hovmoller Plot

```
lim_lat <- range(Tmax_long$lat)  # latitude range
lim_t <- range(Tmax_long$t) # time range
lat_axis <- seq(lim_lat[1], # latitude axis
                lim_lat[2],
                length=25)
t_axis <- seq(lim_t[1], # time axis
              lim_t[2],
              length=100)
lat_t_grid <- expand.grid(lat = lat_axis,
                          t = t_axis)
```

```
Tmax_grid <- Tmax_long
dists <- abs(outer(Tmax_long$lat, lat_axis, "-"))
Tmax_grid$lat <- lat_axis[apply(dists, 1, which.min)]
```

```
 Tmax_lat_Hov <- Tmax_grid %>% group_by(lat, t) %>%
  summarise(z = mean(values))
```

```
ggplot(Tmax_lat_Hov) + # take data
  geom_tile(aes(x = lat, y = t, fill = z)) + # plot
  fill_scale(name = "degF") +
  scale_y_reverse() +
  ylab("Day number (days)") +
  xlab("Latitude (degrees)") +
  theme_bw()
```



On your own

For the SST data loaded using the code below:

1. generate a data frame with the Empirical Spatial Means per decade (1970-1979, 1980-1989, 1990-2002) and plot them with one panel per decade

2. generate a spatial plot for the **yearly** SST 95th quantile for the years 1980, 1990, 2000 having one panel per year

3. Obtain a Hovmoller plot for these data

4. Calculate the EOFs for the *SST* dataset. Replicate the figures in pages 44-45 using the R function `eigen` (not `svd`) and `ggplot2::geom\_tile` and interpret them. How many EOFs would you retain?

```
data("SSTlandmask", package = "STRbook")
data("SSTlonlat", package = "STRbook")
data("SSTdata", package = "STRbook")

#remove years that are not complete
rm_rows <- which(SSTlandmask == 1)
SSTdata <- SSTdata[-rm_rows, 1:396]
```