BST 6200 Spatial Statistics and Disease Mapping
Homework 4

Miao Cai

Overall Goal: Perform an ecological study of the relationship between smoking and pancreatic cancer rates in the state of Minnesota.

1 Homework description

Obtain the shape file for Minnesota through the tigris package.

Obtain the smoking data by selecting Minnesota and Adult Smoking here:

https://www.countyhealthrankings.org/explore-health-rankings/measures-data-sources/county-health-rankings-model/health-factors/health-behaviors/tobacco-use

You can probably copy the data and paste it into Excel.

Obtain the number of cases and the population size here: https://data.web.health.state.mn.us/cancer_query.

Use shift-click to select all counties. Then select “Pancreas” under Indicator, “2012-2016” under Year, and “All” under Sex. Then click “Submit” and finally “Download” at the bottom of the data. Two of the columns in the resulting file are “count” and “population”. You will need both of these. You might want to clean up this file using Excel.

As part of your report, give choropleth maps of smoking rates and pancreatic cancer rates. Compute Moran’s I for both and assess their significance.

Run a regression model of the form:

$$\begin{aligned}
&Y_i \sim \text{Poisson}(P_i\eta_i)\\
\text{where}&\\
&\eta_i = \exp(\beta_0 + \beta_1 x_i + u_i + \nu_i)\\
&P_i = \text{population of county } i\\
&u_i \text{ is correlated heterogeneity of county } i\\
&v_i \text{ is uncorrelated heterogeity of county } i\\
&x_i \text{ is the smoking rate of county } i
\end{aligned}$$

Write a report addressing the question of smoking and pancreatic cancer. The project shouldn’t be too long. I’m thinking something like 3 to 6 pages, counting figures.

[Note: This project is much like the term project for the course, except I have pointed you to the data and given you a specific model with just one predictor variable (smoking). You should look for more than one predictor variable in your term project model.]

2 Data

This part shows the R code to read Minnesota `.shp` file from the `tigris` package, pancreas cancer and adult smoking rates data, and data cleaning and merging process.

```
pacman::p_load(tigris, sp, spdep, rgeos, sf, tmap,
               tidyverse, data.table, nimble)
MN = counties("Minnesota", cb = TRUE)
```

```
##   |                                                                         |
```

```
adult_smoke = fread("data/Minnesota_adult_smoking.csv") %>%
  dplyr::select(FIPS, smoking_rate = `% Smokers`)
pancreas = fread("data/Minnesota_Pancreas_cancer.csv") %>%
  dplyr::select(FIPS = fips, year, N_pancreas = count,
         population, pancreas_rate = rate) %>%
  mutate(pancreas_rate = gsub(" \\(UR\\)", "", pancreas_rate) %>%
           as.numeric())

MN_data = MN@data %>%
  mutate(FIPS = as.integer(paste0(STATEFP, COUNTYFP))) %>%
  dplyr::select(NAME, FIPS) %>%
  left_join(pancreas, by = 'FIPS') %>%
  left_join(adult_smoke, by = 'FIPS')

MN@data = MN_data
```

3 Choropleth maps

```
map_pancreas = tm_shape(MN) +
  tm_fill(title = "Pancreas cancer\nrates",
          col = "pancreas_rate",
          n = 6, style = "jenks",
          palette = "Reds") +
  tm_borders(col = "black") +
  tm_layout(main.title = "Pancreas cancer rates in Minnesota",
            main.title.size = 1.2, frame = FALSE) +
  tm_legend(legend.position = c(0.67, 0.2))

map_smoking = tm_shape(MN) +
  tm_fill(title = "Adult smoking\nrates",
          col = "smoking_rate",
          n = 6, style = "jenks",
          palette = "Reds") +
  tm_borders(col = "black") +
  tm_layout(main.title = "Adult smoking rates in Minnesota",
            main.title.size = 1.2, frame = FALSE) +
  tm_legend(legend.position = c(0.67, 0.2))

tmap_arrange(map_pancreas, map_smoking, ncol = 2)
```

Based on visual inspection of the two choropleth maps, it seems that adult smoking rates have a pretty clear pattern of spatial clustering, while pancreas cancer rates do not have such an obvious trend.

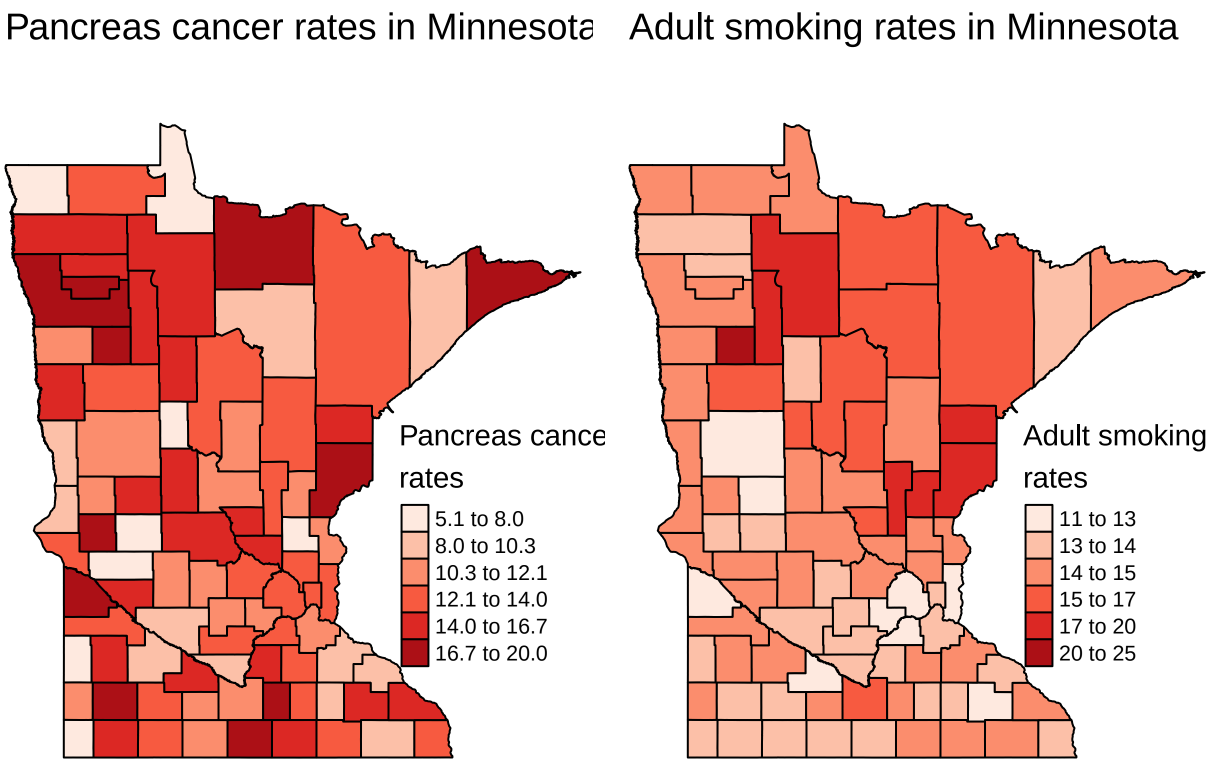


Figure 1: Choropleth maps of pancreas cancer rates (left) and adult smoking rates (right) in Minnesota

4 Moran's I

```
MN_nb = spdep::poly2nb(MN, queen = T)
MN_weight = spdep::nb2listw(MN_nb, style = 'W', zero.policy = TRUE)
```

```
spdep::moran.test(MN$pancreas_rate, MN_weight)
```

```
##
##  Moran I test under randomisation
##
## data:  MN$pancreas_rate
## weights: MN_weight
##
## Moran I statistic standard deviate = -0.57475, p-value = 0.7173
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      -0.050871103      -0.011627907       0.004662061
```

I find that the Moran's I of pancreas cancer rate is very close to 0 and not statistically significant, which suggest that there is no significant spatial clustering effect for county-level pancreas cancer rates in Minnesota.

```
spdep::moran.test(MN$smoking_rate, MN_weight)
```

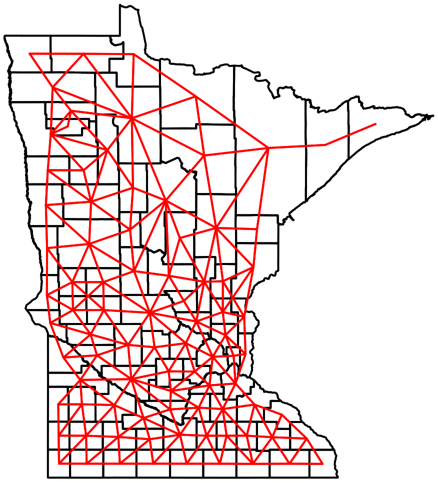
```
##
##  Moran I test under randomisation
##
## data:  MN$smoking_rate
## weights: MN_weight
##
## Moran I statistic standard deviate = 4.8233, p-value = 7.06e-07
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##        0.29996002       -0.01162791        0.00417324
```

The Moran's I of adult smoking rate is around 0.3 and statistically significant, which suggest that there is significant spatial clustering effect for county-level adult smoking rates in Minnesota.

5 Regression modeling

```
MN_nb = spdep::poly2nb(MN, queen = T)
MN_net = spdep::nb2lines(MN_nb , coords = coordinates(MN))
MN_lw = spdep::nb2listw( MN_nb )

plot(MN)
plot(MN_nb, coordinates(MN), add = TRUE, points = FALSE, pch = 16, col = 'red')
```



```
k = nrow(MN@data)
num = rep(0, k)
for (i in 1:k) num[i] = length( MN_lw$neighbours[[i]] )
adj = c()
for (i in 1:k) adj = c(adj, MN_lw$neighbours[[i]] )
L = length(adj)

code = nimbleCode({
  mu ~ dnorm(0, sd = 5)
  tau ~ dgamma(1 , 0.001)
  tau1 ~ dgamma(1 , 0.001)
  beta ~ dnorm(0, sd = 5)
  for (i in 1:L)
    weights[i]  <-  1
  s[1:k] ~ dcar_normal(adj[1:L], weights[1:L], num[1:k], tau, zero_mean = 1)
  for (i in 1:k) {
    log(theta[i])  <-  mu + beta * x[i] + s[i] + v[i]
    y[i] ~ dpois(population[i] * theta[i])
    v[i] ~ dnorm(0 , tau1)
  }
})

constants = list(
  k = k,
  L = L,
  num = num,
  adj = adj,
  population = MN$population
)
data = list(y = MN$N_pancreas, x = MN$smoking_rate)
inits = list(mu = 0, tau = 1, s = rep(0, k), beta = 0, tau1 = 1)

MN_model = nimbleModel(
  code = code ,
  constants = constants ,
  data = data ,
  inits = inits
)
```

```
## defining model...

## Adding num, adj as data for building model.

## building model...

## setting data and initial values...

## running calculate on model (any error reports that follow may simply reflect missing values in model
## checking model sizes and dimensions... This model is not fully initialized. This is not an error. To
## model building finished.
```

```
compiled_MN_model = compileNimble( MN_model )
```

```
## compiling... this may take a minute. Use 'showCompilerOutput = TRUE' to see C++ compilation details.
## compilation finished.
```

```
MN_Conf = configureMCMC( MN_model , print = TRUE )
```

```
## ===== Monitors =====
## thin = 1: mu, tau, tau1, beta
## ===== Samplers =====
## RW sampler (90)
##   - mu
##   - tau
##   - beta
##   - v[]  (87 elements)
## conjugate sampler (1)
##   - tau1
## CAR_normal sampler (1)
##   - s[1:87]
```

```
MN_Conf$addMonitors(c("mu","tau","theta", "beta", "tau1"))
```

```
## thin = 1: mu, tau, tau1, beta, theta
```

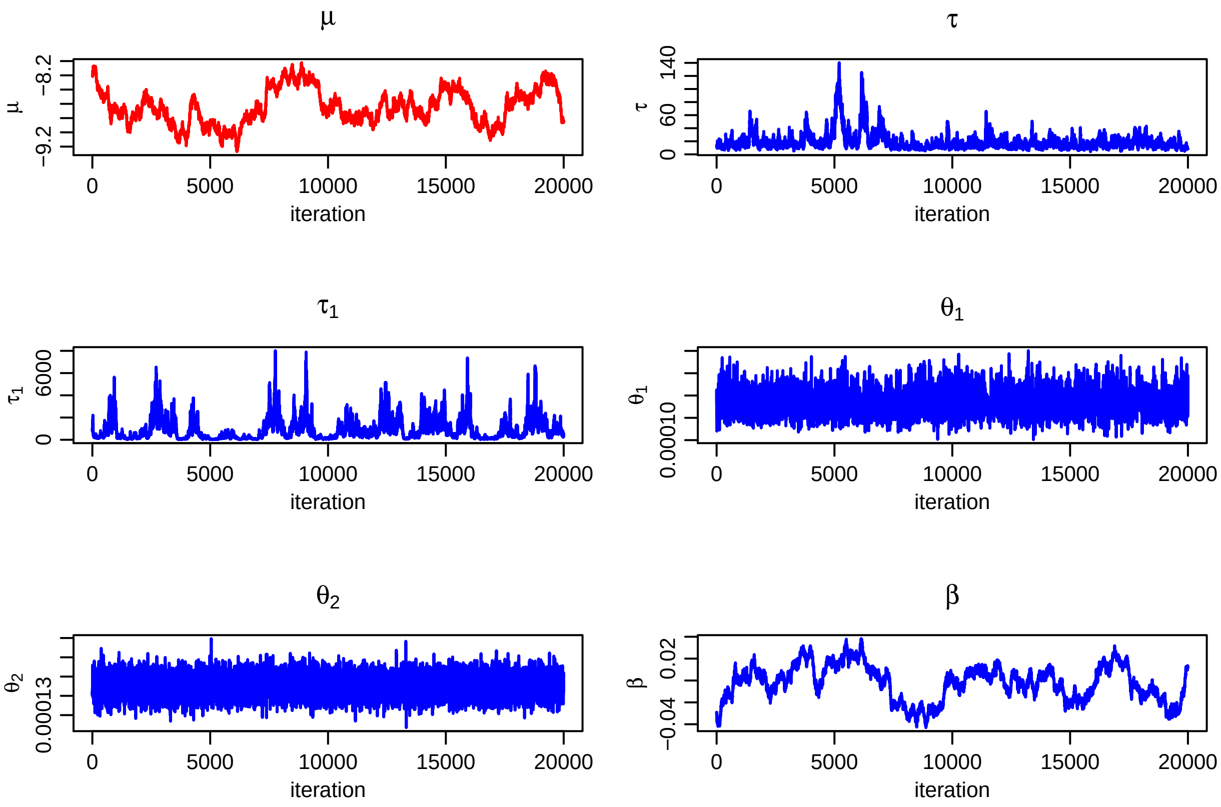
```
MN_Conf_MCMC = buildMCMC( MN_Conf )
MN_complie_MCMC = compileNimble( MN_Conf_MCMC )
```

```
## compiling... this may take a minute. Use 'showCompilerOutput = TRUE' to see C++ compilation details.
## compilation finished.
```

```
set.seed(1)
MCMC_samples = runMCMC(MN_complie_MCMC, niter = 40000, nburnin = 20000,
                   inits = inits, nchains = 1, samplesAsCodaMCMC = TRUE)
```

```
## running chain 1...
## |-------------|-------------|-------------|-------------|
## |-------------------------------------------------------|
```

```
par(mfrow = c(3, 2), mai = c(.6, .5, .4, .1), mgp = c(1.8, 0.7, 0))
ts.plot(MCMC_samples[ , 'mu'], xlab = 'iteration', col="red" , lwd=1.5 ,
        ylab = expression(mu), main = expression(mu))
ts.plot(MCMC_samples[ , 'tau'], xlab = 'iteration', col="blue" , lwd=1.5 ,
        ylab = expression(tau), main = expression(tau))
ts.plot(MCMC_samples[ , 'tau1'], xlab = 'iteration', col="blue" , lwd=1.5 ,
        ylab = expression(tau[1]), main = expression(tau[1]))
ts.plot(MCMC_samples[ , 'theta[1]'], xlab = 'iteration', col="blue" , lwd=1.5 ,
        ylab = expression(theta[1]), main = expression(theta[1]))
ts.plot(MCMC_samples[ , 'theta[2]'], xlab = 'iteration', col="blue" , lwd=1.5 ,
        ylab = expression(theta[2]), main = expression(theta[2]))
ts.plot(MCMC_samples[ , 'beta'], xlab = 'iteration', col="blue" , lwd=1.5 ,
        ylab = expression(beta), main = expression(beta))
```



```
source('code/DBDA2E-utilities.R')
```

```
##
## *********************************************************************
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
## *********************************************************************

## Warning: package 'rjags' was built under R version 3.6.3

## Loading required package: coda
## Linked to JAGS 4.3.0
```

```
## Loaded modules: basemod,bugs
```

```
## Warning: package 'runjags' was built under R version 3.6.3
```

```
##
## Attaching package: 'runjags'
```

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

```
par(mfrow = c(2, 2), mai = c(.6, .5, .4, .1), mgp = c(1.8, 0.7, 0))
plotPost(MCMC_samples[ , 'mu'], main = expression(mu))
```

```
##                   ESS      mean    median      mode hdiMass    hdiLow
## Param. Val. 13.57476 -8.641478 -8.666124 -8.748404    0.95 -9.065969
##               hdiHigh compVal pGtCompVal ROPElow ROPEhigh pLtROPE pInROPE
## Param. Val. -8.179192      NA         NA      NA       NA      NA      NA
##             pGtROPE
## Param. Val.      NA
```

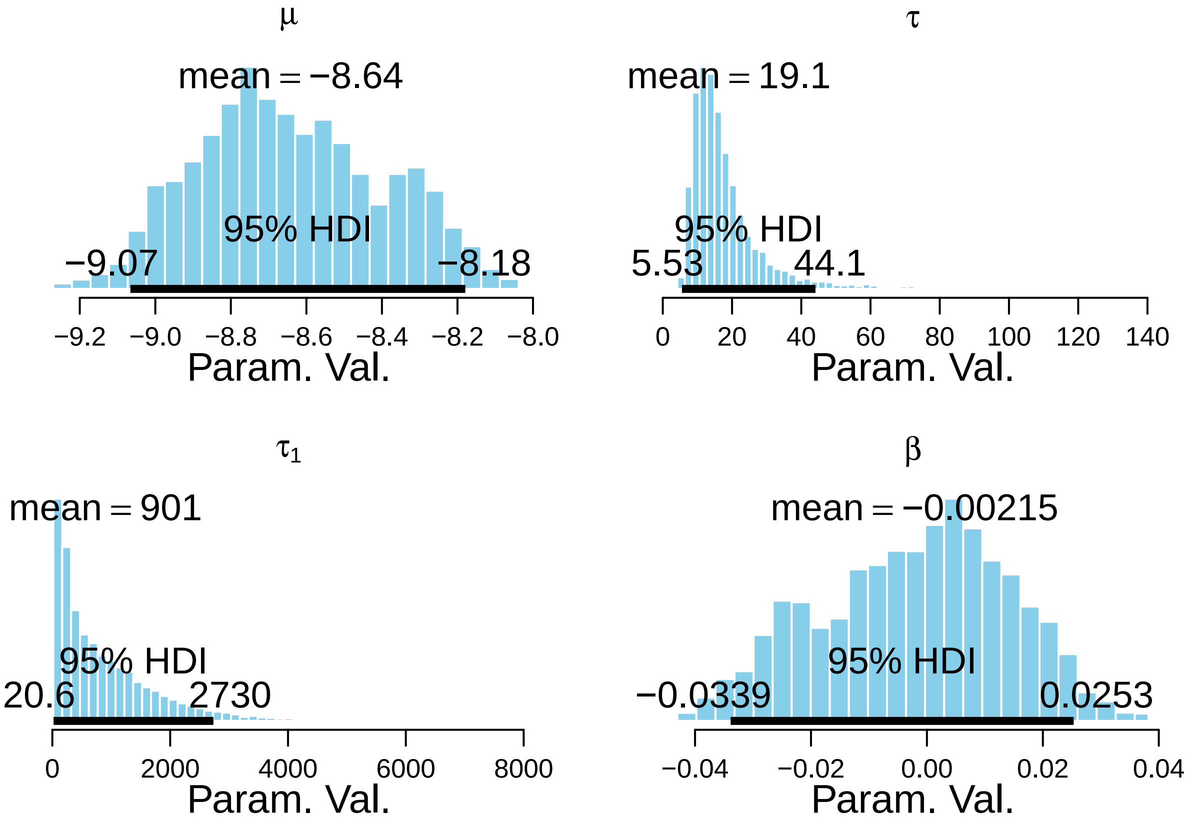
```
plotPost(MCMC_samples[ , 'tau'], main = expression(tau))
```

```
##                   ESS     mean   median     mode hdiMass   hdiLow  hdiHigh
## Param. Val. 64.36702 19.09255 15.22209 12.55553    0.95 5.525563 44.09733
##             compVal pGtCompVal ROPElow ROPEhigh pLtROPE pInROPE pGtROPE
## Param. Val.      NA         NA      NA       NA      NA      NA      NA
```

```
plotPost(MCMC_samples[ , 'tau1'], main = expression(tau[1]))
```

```
##                   ESS     mean   median     mode hdiMass   hdiLow hdiHigh
## Param. Val. 94.45487 901.0303 599.6307 174.4559    0.95 20.62201 2734.05
##             compVal pGtCompVal ROPElow ROPEhigh pLtROPE pInROPE pGtROPE
## Param. Val.      NA         NA      NA       NA      NA      NA      NA
```

```
plotPost(MCMC_samples[ , 'beta'], main = expression(beta))
```



```
##                   ESS         mean        median         mode hdiMass
## Param. Val. 12.85135 -0.002145628 -0.0005530854 0.005161487    0.95
##                   hdiLow    hdiHigh compVal pGtCompVal ROPElow ROPEhigh
## Param. Val. -0.03385938 0.02531968      NA         NA      NA       NA
##             pLtROPE pInROPE pGtROPE
## Param. Val.      NA      NA      NA
```

The posterior mean of the parameter β is -0.00215 (95% credible interval: $[-0.0339, -0.253]$). Since the 95% credible interval covers 0, it suggests that adult smoking rate does not have significant effects on pancreas cancer rates in our model.

```r
sessionInfo()
```

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 16299)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] runjags_2.0.4-6    rjags_4-10        coda_0.19-2
##  [4] nimble_0.9.0       data.table_1.12.8 forcats_0.4.0
##  [7] stringr_1.4.0      dplyr_0.8.3       purrr_0.3.3
## [10] readr_1.3.1        tidyr_1.0.0       tibble_2.1.3
## [13] ggplot2_3.3.0.9000 tidyverse_1.2.1   tmap_2.3-2
## [16] rgeos_0.5-2        spdep_1.1-2       sf_0.8-0
## [19] spData_0.3.0       sp_1.3-1          tigris_0.9.4
##
## loaded via a namespace (and not attached):
##  [1] colorspace_1.4-1   deldir_0.1-16      class_7.3-15
##  [4] leaflet_2.0.2      rgdal_1.4-7        dichromat_2.0-0
##  [7] rstudioapi_0.11    fansi_0.4.1        lubridate_1.7.4
## [10] xml2_1.2.2         codetools_0.2-16   splines_3.6.2
## [13] knitr_1.22         jsonlite_1.6.1     tmaptools_2.0-2
## [16] broom_0.5.2        shiny_1.4.0        compiler_3.6.2
## [19] httr_1.4.0         backports_1.1.5    assertthat_0.2.1
## [22] Matrix_1.2-18      fastmap_1.0.1      cli_2.0.1
## [25] later_1.0.0        htmltools_0.4.0    tools_3.6.2
## [28] igraph_1.2.4.1     gtable_0.3.0       glue_1.3.1
## [31] rappdirs_0.3.1     gmodels_2.18.1     Rcpp_1.0.3
## [34] cellranger_1.1.0   raster_3.0-7       vctrs_0.2.2
## [37] gdata_2.18.0       nlme_3.1-142       leafsync_0.1.0
## [40] crosstalk_1.0.0    lwgeom_0.2-1       xfun_0.12
## [43] rvest_0.3.3        mime_0.9           lifecycle_0.1.0
## [46] pacman_0.5.1       gtools_3.8.1       XML_3.98-1.19
## [49] LearnBayes_2.15.1  MASS_7.3-51.4      scales_1.1.0
## [52] hms_0.4.2          promises_1.1.0     expm_0.999-4
## [55] RColorBrewer_1.1-2 yaml_2.2.0         curl_4.2
## [58] stringi_1.4.5      highr_0.8          maptools_0.9-5
## [61] e1071_1.7-2        boot_1.3-23        rlang_0.4.4
## [64] pkgconfig_2.0.3    evaluate_0.14      lattice_0.20-38
## [67] htmlwidgets_1.5.1  tidyselect_0.2.5   magrittr_1.5
## [70] R6_2.4.1           generics_0.0.2     DBI_1.0.0
## [73] pillar_1.4.3       haven_2.1.0        foreign_0.8-72
## [76] withr_2.1.2        units_0.6-5        modelr_0.1.4
## [79] crayon_1.3.4       uuid_0.1-4         KernSmooth_2.23-16
## [82] rmarkdown_2.1      grid_3.6.2         readxl_1.3.1
## [85] digest_0.6.24      classInt_0.4-2     xtable_1.8-4
## [88] httpuv_1.5.2       munsell_0.5.0      viridisLite_0.3.0
```