NYPD Shooting Data

2024-03-22

Importing Data and Libraries

The first few cells will be importing and cleaning the NYPD Historical Shooting Data into R. We also will load all our packages for use throughout the entire script.

```
library(tidyr)
library(ggplot2)
library(dplyr)
library(rnaturalearth)
library(rnaturalearthdata)
library(viridis)
library(RCurl)
```

```
x <- getURL("https://raw.githubusercontent.com/alec-sekelsky/NYPD-Shooting-Data/main/NYPD_Shooting_Inci
nypd <- read.csv(text = x)

summary(nypd)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME            BORO
##  Min.   :  9953245   Length:27312       Length:27312       Length:27312
##  1st Qu.: 63860880   Class :character   Class :character   Class :character
##  Median : 90372218   Mode  :character   Mode  :character   Mode  :character
##  Mean   :120860536
##  3rd Qu.:188810230
##  Max.   :261190187
##
##  LOC_OF_OCCUR_DESC     PRECINCT      JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:27312       Min.   :  1.00   Min.   :0.0000    Length:27312
##  Class :character   1st Qu.: 44.00   1st Qu.:0.0000    Class :character
##  Mode  :character   Median : 68.00   Median :0.0000    Mode  :character
##                     Mean   : 65.64   Mean   :0.3269
##                     3rd Qu.: 81.00   3rd Qu.:0.0000
##                     Max.   :123.00   Max.   :2.0000
##                                      NA’s   :2
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  Length:27312       Length:27312            Length:27312
##  Class :character   Class :character        Class :character
##  Mode  :character   Mode  :character        Mode  :character
##
##
##
##
##    PERP_SEX           PERP_RACE          VIC_AGE_GROUP        VIC_SEX
##  Length:27312       Length:27312       Length:27312       Length:27312
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_RACE           X_COORD_CD        Y_COORD_CD        Latitude
##  Length:27312       Min.   : 914928   Min.   :125757   Min.   :40.51
##  Class :character   1st Qu.:1000028   1st Qu.:182834   1st Qu.:40.67
##  Mode  :character   Median :1007731   Median :194487   Median :40.70
##                     Mean   :1009449   Mean   :208127   Mean   :40.74
##                     3rd Qu.:1016838   3rd Qu.:239518   3rd Qu.:40.82
##                     Max.   :1066815   Max.   :271128   Max.   :40.91
##                                                        NA’s   :10
##    Longitude        Lon_Lat
##  Min.   :-74.25   Length:27312
##  1st Qu.:-73.94   Class :character
##  Median :-73.92   Mode  :character
##  Mean   :-73.91
##  3rd Qu.:-73.88
##  Max.   :-73.70
##  NA’s   :10
```

Basic Cleaning of the Data

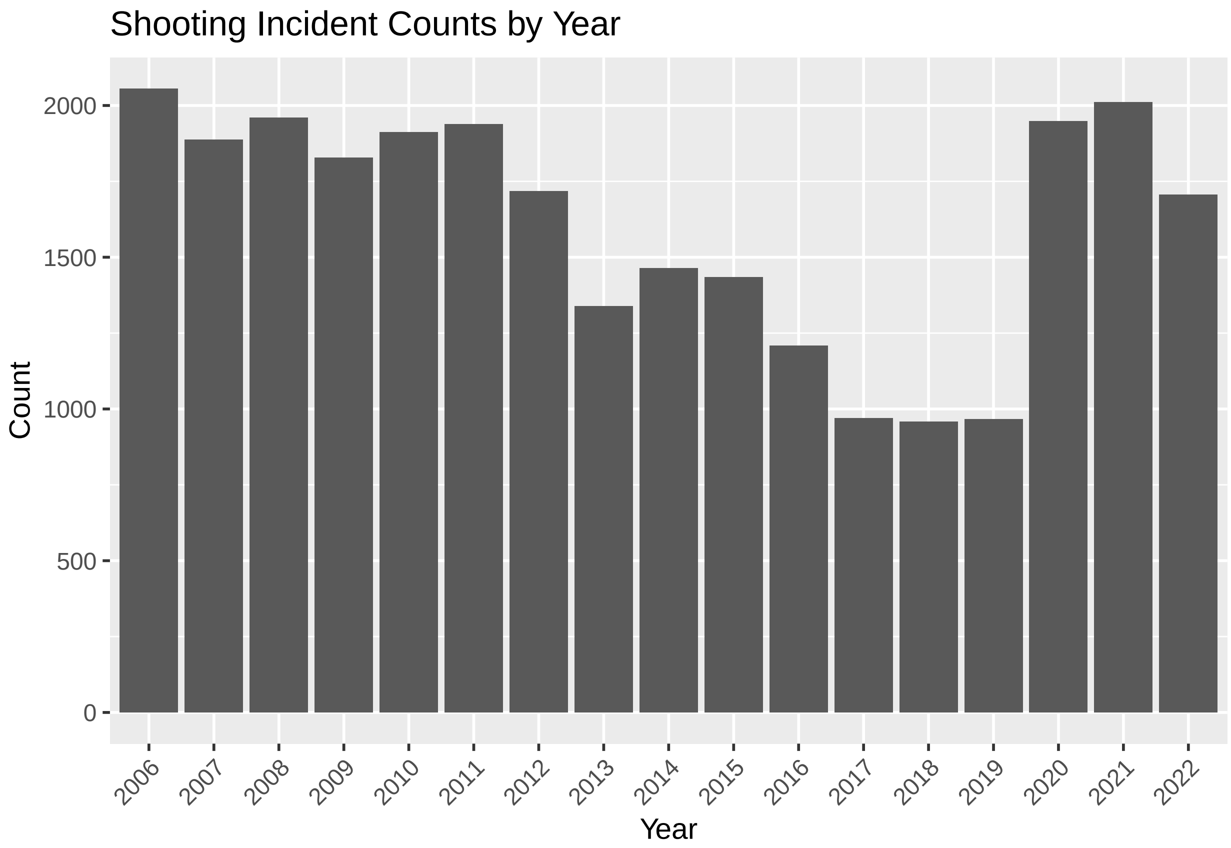
From a glance at the data, we can see some columns that may be irrelevant for a simple analyis. Headers like jurisdiction code, LOC_CLASSFCTN_DESC, X_COORD_CD, Y_COORD_CD, Latitude, Longitude, and Lon_lat will most likely be removed. Latitude and Longitude also have several NA values which would not be worth much to us. There are a few others like PERP_SEX, PERP_AGE_GROUP, PERP_RACE may be removed, but could be useful. There are a lot of missing data points in those columns rendering them mostly unuseful. This is a very clean data set making our job pretty easy.

```
nypd_sub <- subset(nypd, select = -c(JURISDICTION_CODE, LOC_CLASSFCTN_DESC, LOCATION_DESC, LOC_OF_OCCUR_
nypd_sub <- nypd_sub[complete.cases(nypd_sub[]),]
summary(nypd_sub)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME            BORO
##  Min.   :  9953245   Length:27302       Length:27302       Length:27302
##  1st Qu.: 63859932   Class :character   Class :character   Class :character
##  Median : 90340495   Mode  :character   Mode  :character   Mode  :character
##  Mean   :120812265
##  3rd Qu.:188610564
##  Max.   :261190187
##     PRECINCT      STATISTICAL_MURDER_FLAG VIC_AGE_GROUP        VIC_SEX
##  Min.   :  1.00   Length:27302            Length:27302       Length:27302
##  1st Qu.: 44.00   Class :character        Class :character   Class :character
##  Median : 68.00   Mode  :character        Mode  :character   Mode  :character
##  Mean   : 65.64
##  3rd Qu.: 81.00
##  Max.   :123.00
##    VIC_RACE            Latitude       Longitude        Lon_Lat
##  Length:27302       Min.   :40.51   Min.   :-74.25   Length:27302
##  Class :character   1st Qu.:40.67   1st Qu.:-73.94   Class :character
##  Mode  :character   Median :40.70   Median :-73.92   Mode  :character
##                     Mean   :40.74   Mean   :-73.91
##                     3rd Qu.:40.82   3rd Qu.:-73.88
##                     Max.   :40.91   Max.   :-73.70
```

Visualizing the Data

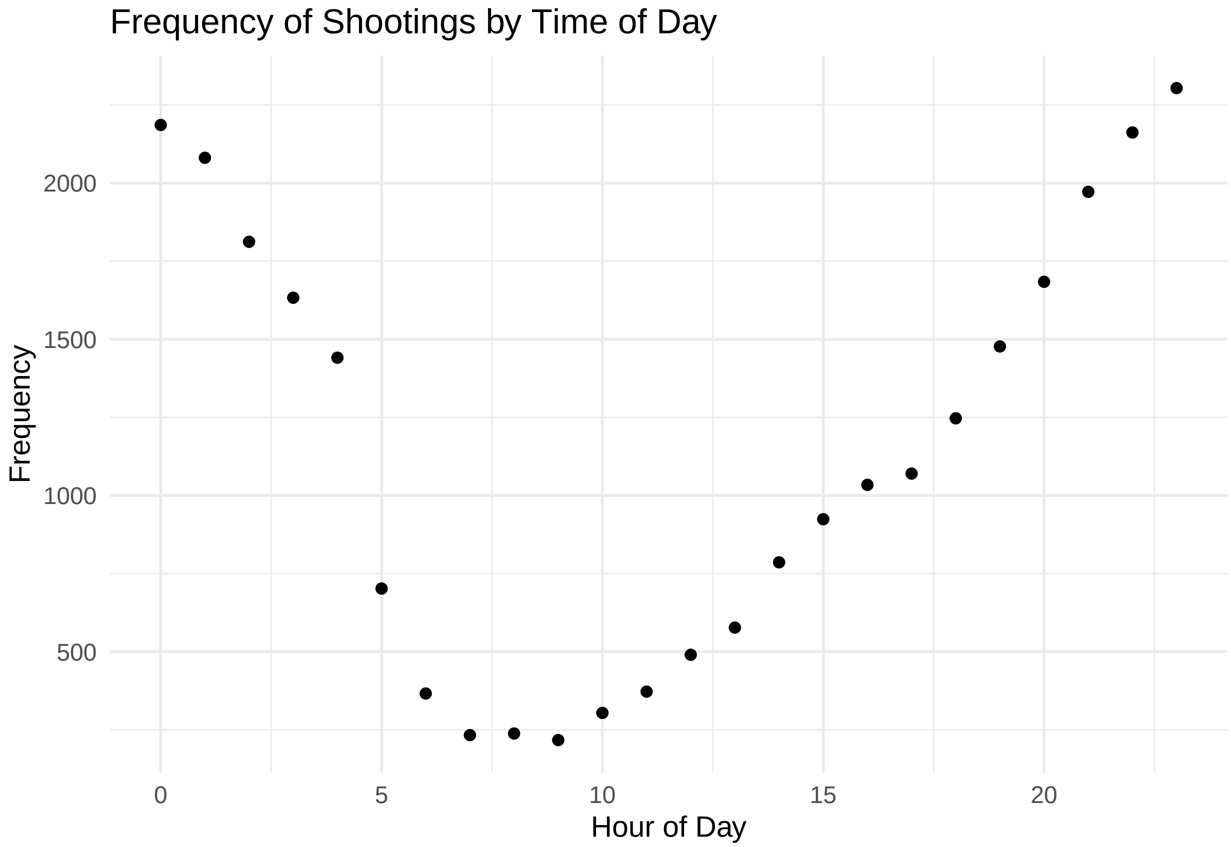
```
nypd_sub$OCCUR_DATE <- as.Date(nypd_sub$OCCUR_DATE, format = "%m/%d/%Y")
nypd_sub$Year <- format(nypd_sub$OCCUR_DATE, "%Y")

ggplot(nypd_sub, aes(x = Year)) +
  geom_bar() +
  labs(title = "Shooting Incident Counts by Year",
       x = "Year",
       y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



This first chart shows the shooting incidents grouped in a bar chart by year. I find it interesting that total shootings were in a decline until 2020 and then shot up by almost 1000. You would think that with lockdowns in place for the 2020 COVID Pandemic we would see a decline.

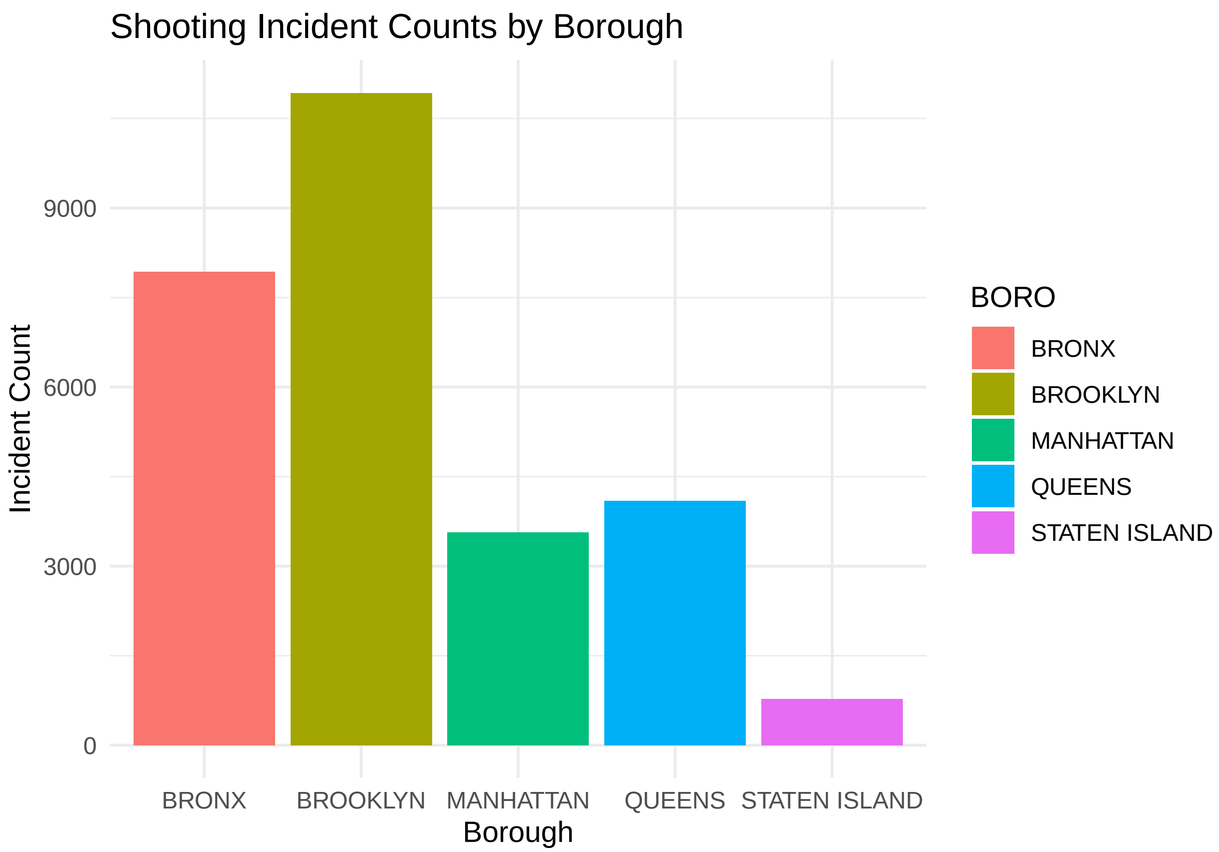
```
nypd$OCCUR_TIME <- as.POSIXct(strptime(nypd$OCCUR_TIME, format = "%H:%M:%S"))
```

```
nypd$Hour <- as.numeric(format(nypd$OCCUR_TIME, "%H"))

hourly_counts <- table(nypd$Hour)

hourly_counts_df <- data.frame(Hour = as.numeric(names(hourly_counts)), Frequency = as.numeric(hourly_c

ggplot(hourly_counts_df, aes(x = Hour, y = Frequency)) +
  geom_point() +
  labs(title = "Frequency of Shootings by Time of Day",
       x = "Hour of Day",
       y = "Frequency") +
  theme_minimal()
```



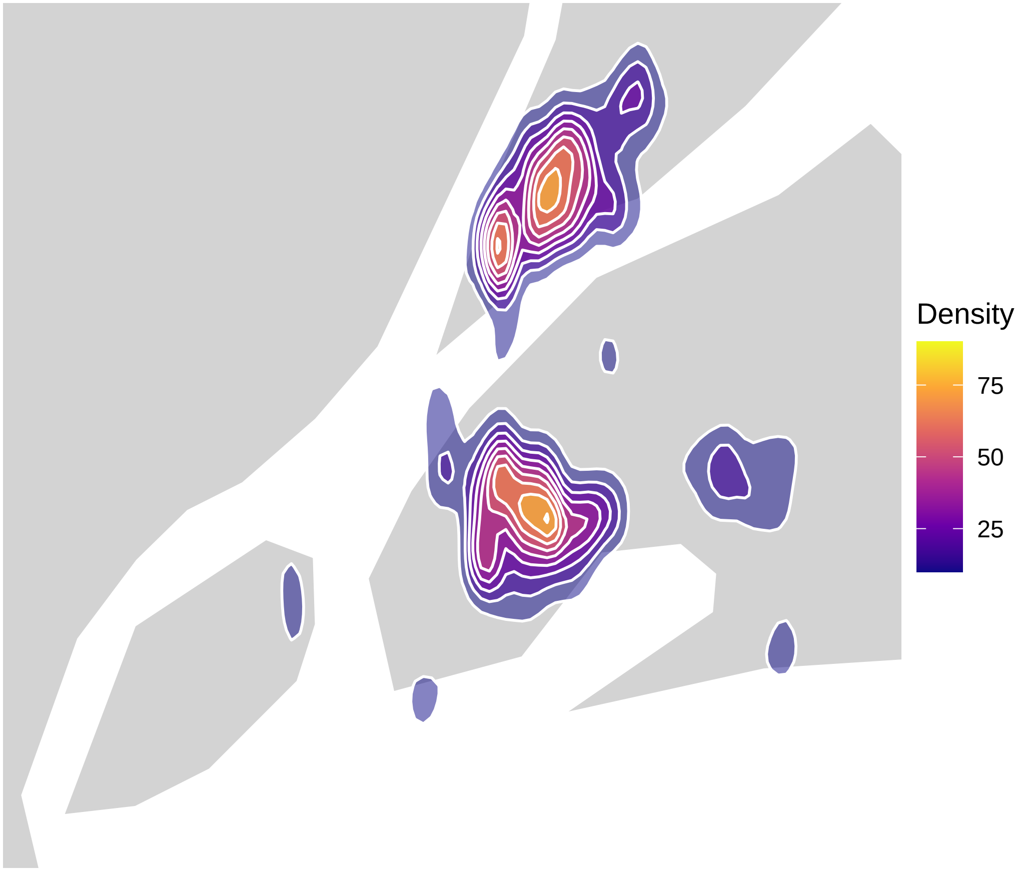
This second plot shows frequency of shootings compared to time of day. We can infer from this chart that as the day goes on there is more of a likelihood of a shooting occuring during nighttime hours.

```
nypd_sub %>%
  group_by(BORO) %>%
  summarise(incident_count = n()) %>%
  ggplot(aes(x = BORO, y = incident_count, fill = BORO)) +
  geom_bar(stat = "identity") +
  labs(title = "Shooting Incident Counts by Borough",
       x = "Borough",
       y = "Incident Count") +
  theme_minimal()
```



This chart shows total shootings by Borough. This chart gives a brief insight into boroughs that can be inferred as more dangerous or violent. I would like to dive deeper into this analysis in the future. More data can be used to supplement this and possibly give some leads into why we see more violent crime in these boroughs.

```r
world_map <- ne_countries(scale = "medium", returnclass = "sf")

map <- ggplot() +
  geom_sf(data = world_map, fill = "lightgray", color = "white") +
  coord_sf(xlim = range(nypd_sub$Longitude), ylim = range(nypd_sub$Latitude)) +
  theme_void()

map +
  stat_density_2d(data = nypd_sub, aes(x = Longitude, y = Latitude, fill = after_stat(level)),
                  geom = "polygon", color = "white", alpha = 0.5) +
  scale_fill_viridis_c(option = "plasma", name = "Density") +
  theme_void()
```



This last chart shows a desnity plot of shootings and where they occur. It backs up the bar chart above showing that Queens, Brooklyn, and the Bronx are the most frequent areas of a shooting occuring.

Data Model

```
nypd_mod_sub = subset(nypd, select = c(STATISTICAL_MURDER_FLAG, PRECINCT, X_COORD_CD, Y_COORD_CD))

nypd_mod_sub = na.omit(nypd_mod_sub)
nypd_mod_sub$STATISTICAL_MURDER_FLAG <- as.numeric(nypd_mod_sub$STATISTICAL_MURDER_FLAG == "true")

model = lm(nypd_mod_sub$STATISTICAL_MURDER_FLAG ~ nypd_mod_sub$PRECINCT + nypd_mod_sub$X_COORD_CD + nypd
summary(model)
```

```
##
## Call:
## lm(formula = nypd_mod_sub$STATISTICAL_MURDER_FLAG ~ nypd_mod_sub$PRECINCT +
##     nypd_mod_sub$X_COORD_CD + nypd_mod_sub$Y_COORD_CD)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2038 -0.1946 -0.1904 -0.1865  0.8190
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.952e-01  1.346e-01   1.450    0.147
## nypd_mod_sub$PRECINCT    1.867e-04  1.241e-04   1.504    0.133
## nypd_mod_sub$X_COORD_CD -1.923e-08  1.416e-07  -0.136    0.892
## nypd_mod_sub$Y_COORD_CD  2.289e-08  1.053e-07   0.217    0.828
##
## Residual standard error: 0.3945 on 27308 degrees of freedom
## Multiple R-squared:  0.0001352,  Adjusted R-squared:  2.536e-05
## F-statistic: 1.231 on 3 and 27308 DF,  p-value: 0.2966
```

Looking at the summary of this model, we can tell its a very poor model. With an R^2 of 0.00014 and a p-value of 0.2966 there is much to improve on future models. Using this as a predictor for where a murder might of occured is not something I would do.

Potential Bias

The biggest thing that stands out to me in terms of Bias when analyzing this data is the assumptions we may make about our conclusions. In my second graph, I showed NYPD shootings by Borough. Brooklyn showed as the most frequent Borough for shootings, but why? Was there actually an uptick of crime or violence in that area requiring officers using lethal force or is there another reason? Maybe the training is more poor there or there are less officers and they are put in more dangerous situations. We would need to have some amplifing data here to confirm our bias.

We should also consider population of a borough, i.e. a borough with a lower population may have a lower freuquncy of shootings than a borough with a much larger population.

```
sessionInfo()
```

```
## R version 4.3.3 (2024-02-29)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Ventura 13.5.2
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  LAPACK ve
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] RCurl_1.98-1.14         viridis_0.6.5           viridisLite_0.4.2
## [4] rnaturalearthdata_1.0.0 rnaturalearth_1.0.1     dplyr_1.1.4
## [7] ggplot2_3.5.0           tidyr_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4         generics_0.1.3     bitops_1.0-7       class_7.3-22
##  [5] KernSmooth_2.23-22 digest_0.6.35      magrittr_2.0.3     evaluate_0.23
##  [9] grid_4.3.3         fastmap_1.1.1      jsonlite_1.8.8     e1071_1.7-14
## [13] DBI_1.2.2          gridExtra_2.3      httr_1.4.7         purrr_1.0.2
## [17] fansi_1.0.6        scales_1.3.0       isoband_0.2.7      codetools_0.2-19
## [21] cli_3.6.2          rlang_1.1.3        units_0.8-5        munsell_0.5.0
## [25] withr_3.0.0        yaml_2.3.8         tools_4.3.3        colorspace_2.1-0
## [29] vctrs_0.6.5        R6_2.5.1           proxy_0.4-27       lifecycle_1.0.4
## [33] classInt_0.4-10    MASS_7.3-60.0.1    pkgconfig_2.0.3    terra_1.7-71
## [37] pillar_1.9.0       gtable_0.3.4       glue_1.7.0         Rcpp_1.0.12
## [41] sf_1.0-16          highr_0.10         xfun_0.42          tibble_3.2.1
## [45] tidyselect_1.2.1   rstudioapi_0.15.0  knitr_1.45         farver_2.1.1
## [49] htmltools_0.5.7    labeling_0.4.3     rmarkdown_2.26     compiler_4.3.3
```