```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(cancensus)
```

```
## Census data is currently stored temporarily.
##
##  In order to speed up performance, reduce API quota usage, and reduce unnecessary network calls, plea
##  This will add your cache directory as environment varianble to your .Renviron to be used across sess
```

```r
library(knitr)
library(readr)
library(sf)
```

```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```r
library(geojsonsf)
library(paletteer)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```r
library(broom)
```

Census data

```r
load("API_KEY.rda")

options(cancensus.api_key = api_key)
options(cancensus.cache_path = "cache")

vectors <- c("v_CA21_1",
             "v_CA21_6",
             "v_CA21_449",
             "v_CA21_1040",
             "v_CA21_1085",
             "v_CA21_905")

census_data <- get_census(
  dataset = "CA21",
  regions = list(
```

```
    CSD = "5915022",
    DA = c(
      "59154012",
      "59154105",
      "59154090",
      "59150936",
      "59154101",
      "59154104",
      "59154035",
      "59154103",
      "59154102",
      "59154034",
      "59150945",
      "59154091",
      "59154093",
      "59154099",
      "59150946",
      "59154100",
      "59154078",
      "59154079",
      "59154082",
      "59154081",
      "59154080",
      "59150939",
      "59150938",
      "59154083",
      "59154095",
      "59154084",
      "59150941",
      "59150942",
      "59154085",
      "59154088",
      "59154087",
      "59154089",
      "59154097",
      "59154098",
      "59154096",
      "59154092",
      "59154013",
      "59150952"
    )
  ),
  vectors = vectors,
  labels = "detailed",
  geo_format = "sf",
  level = "DA"
)
```

```
## Reading vectors data from local cache.
```

```
## Reading geo data from local cache.
```

```
can_api_key <- ""
save(census_data, file = "data/census.rda")
```

Food data

```
food_data <- st_read("data/free-and-low-cost-food-programs.shp")  %>%
  select(
    "program_nam",
    "program_sta",
    "meal_cost",
    "local_areas",
    "latitude",
    "longitude",
    "geometry"
  ) %>%
  drop_na("latitude", "longitude") %>%
  # set to wgs 84 as per can census
  st_set_crs(4326)
```

```
## Reading layer `free-and-low-cost-food-programs' from data source
##   `/Users/sid/Documents/ubc classes/2024w1/econ 326/foodprograms-326/data/free-and-low-cost-food-prog
##   using driver `ESRI Shapefile'
## replacing null geometries with empty geometries
## Simple feature collection with 83 features and 25 fields (with 2 geometries empty)
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -123.1821 ymin: 49.20725 xmax: -123.0287 ymax: 49.286
## CRS:           NA
```

```
# Food data processing
food_count <- food_data %>%
  st_set_geometry(NULL) %>%
  group_by(local_areas) %>%
  summarise(count = n(), .groups = "drop")

food_data_count <- food_data %>%
  left_join(food_count, by = "local_areas") %>%
  distinct(local_areas, .keep_all = TRUE) # one row per neighbourhood

combo_food_census <- census_data %>%
  st_join(food_data_count)

census_data_food <- combo_food_census %>%
  mutate(program_count = replace_na(count, 0),
         food_density = program_count / `Shape Area`)

head(census_data_food) %>%
  kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Shape Area	Type	Households	Quality Flags	name	GeoUID	CSD_UID	Population	CT_UID	Dwell
0.2991	DA	266	0	59150307	59150307	5915022	677	9330053.02	
0.1096	DA	218	0	59150308	59150308	5915022	541	9330053.02	
0.1119	DA	282	0	59150309	59150309	5915022	761	9330053.02	
0.1094	DA	389	0	59150310	59150310	5915022	748	9330053.02	
0.0809	DA	187	0	59150311	59150311	5915022	537	9330053.02	

0.0871	DA	201	0	59150312	59150312	5915022	555	9330053.02

Crime data

```
crime <- read_csv("data/crime_data_all_neighborhoods.csv") %>%
  mutate(TYPE = as_factor(TYPE),
         HUNDRED_BLOCK = as_factor(HUNDRED_BLOCK),
         NEIGHBOURHOOD = as_factor(NEIGHBOURHOOD)) %>%
  filter(!is.na(X) & !is.na(Y))
```

```
## Rows: 32202 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): TYPE, HUNDRED_BLOCK, NEIGHBOURHOOD
## dbl (7): YEAR, MONTH, DAY, HOUR, MINUTE, X, Y
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
crime_data <- st_as_sf(crime, coords = c("X", "Y"), crs = "+proj=utm +zone=10") %>%
  st_transform(crs = "+proj=longlat +datum=WGS84")
```

```
intersections <- st_is_within_distance(census_data, crime_data, sparse = FALSE, dist = 5)

crimes_contained <- rowSums(intersections, dims = 1)

mean(crimes_contained)
```

```
## [1] 41.26376
```

```
census_data_crime <- census_data %>%
  cbind(crimes_contained) %>%
  mutate(crime_density = crimes_contained / Shape.Area)

unique_crimes <- unique(crime_data$TYPE)

for (type in unique_crimes) {
  type_data <- crime_data %>% filter(TYPE == type)
  intersections <- st_is_within_distance(census_data, type_data, sparse = FALSE, dist = 5)
  sum <- rowSums(intersections, dims = 1)
  df <- as.data.frame(sum)
  census_data_crime <- census_data_crime %>% cbind(df$sum) %>% rename_with(~ paste0("crimes_", type), di
  print(mean(sum))
}
```

```
## [1] 2.893738
## [1] 1.957306
## [1] 0
## [1] 7.957306
## [1] 0
## [1] 11.78463
## [1] 11.02657
## [1] 2.156546
## [1] 1.250474
## [1] 0.056926
```

```
## [1] 2.180266
```

```
head(census_data_crime) %>%
  kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Shape.Area	Type	Households	Quality.Flags	name	GeoUID	CSD_UID	Population	CT_UID	Dwelli
0.2991	DA	266	0	59150307	59150307	5915022	677	9330053.02	
0.1096	DA	218	0	59150308	59150308	5915022	541	9330053.02	
0.1119	DA	282	0	59150309	59150309	5915022	761	9330053.02	
0.1094	DA	389	0	59150310	59150310	5915022	748	9330053.02	
0.0809	DA	187	0	59150311	59150311	5915022	537	9330053.02	
0.0871	DA	201	0	59150312	59150312	5915022	555	9330053.02	

Training data

Now we merge all of our datasets into one table so we can feed it into our model.

```
training_data <- st_join(census_data_crime, census_data_food)

training_data <- training_data %>%
  rename_with(
    ~ gsub(":.*$", "", .),  # Remove everything after the colon, including the colon
    starts_with("v_CA21")  # Apply only to columns starting with "v_CA21"
  )

median_lico_at <- median(training_data$v_CA21_1085, na.rm = TRUE)

training_data <- training_data %>%
  mutate(
    low_income = ifelse(
      v_CA21_1085 > median_lico_at,
      1,
      0))


tail(training_data) %>%
  kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

	Shape.Area	Type.x	Households.x	Quality.Flags	name.x	GeoUID.x	CSD_UID.x	Population.x	CT
1054.3	0.1436	DA	267	0	59154196	59154196	5915022	774	93
1054.4	0.1436	DA	267	0	59154196	59154196	5915022	774	93
1054.5	0.1436	DA	267	0	59154196	59154196	5915022	774	93
1054.6	0.1436	DA	267	0	59154196	59154196	5915022	774	93
1054.7	0.1436	DA	267	0	59154196	59154196	5915022	774	93
1054.8	0.1436	DA	267	0	59154196	59154196	5915022	774	93

Model

Specification

```
# Specification models
# 1. Model with all variables (including interaction term)
reg_all_vars <- lm(crime_density ~
                     food_density:low_income +
                     food_density +
                     v_CA21_1 +
                     v_CA21_449 +
                     low_income,
                   data = st_set_geometry(training_data, NULL))

# 2. Model without the interaction term
reg_no_interaction <- lm(crime_density ~
                           food_density +
                           v_CA21_1 +
                           v_CA21_449 +
                           low_income,
                         data = st_set_geometry(training_data, NULL))

# 3. Model with only food density, crime density, and low income
reg_food_crime_low_income <- lm(crime_density ~
                                  food_density +
                                  low_income,
                                data = st_set_geometry(training_data, NULL))

# Summary for each specification tested
summary_all_vars <- summary(reg_all_vars)
summary_no_interaction <- summary(reg_no_interaction)
summary_food_crime_low_income <- summary(reg_food_crime_low_income)

model_summaries <- list(
  "All Variables" = summary_all_vars,
  "Without Interaction" = summary_no_interaction,
  "Food Density, Crime, Low Income" = summary_food_crime_low_income
)
```

RESULTS FROM SPEC

```
for (model_name in names(model_summaries)) {
  cat("\n\n", model_name, "\n")

  # Convert the coefficients to a data frame for display with kable
  coefficients_df <- as.data.frame(model_summaries[[model_name]]$coefficients)

  # Use kable to display the coefficients in a simple table format
  print(kable(coefficients_df, col.names = c("Estimate", "Std. Error", "t value", "Pr(>|t|)")))
}
```

```
##
##
```

All Variables
##
##
| | Estimate| Std. Error| t value| Pr(>|t|)|
|:------------------------|----------:|----------:|----------:|------------------:|
|(Intercept) | 24.088259| 27.1756280| 0.886392| 0.3754341|
|food_density | 2.887305| 2.3284161| 1.240029| 0.2150024|
|v_CA21_1 | 7.611303| 0.2809919| 27.087267| 0.0000000|
|v_CA21_449 | -7.165067| 0.2860258| -25.050420| 0.0000000|
|low_income | 505.044208| 26.1606293| 19.305507| 0.0000000|
|food_density:low_income | 2.608956| 2.5036654| 1.042055| 0.2974192|
##
##
Without Interaction
##
##
| | Estimate| Std. Error| t value| Pr(>|t|)|
|:------------|----------:|----------:|-----------:|------------------:|
|(Intercept) | 22.144163| 27.1116648| 0.8167762| 0.4140816|
|food_density | 5.145253| 0.8522934| 6.0369505| 0.0000000|
|v_CA21_1 | 7.639805| 0.2796590| 27.3182847| 0.0000000|
|v_CA21_449 | -7.193525| 0.2847207| -25.2652007| 0.0000000|
|low_income | 507.066218| 26.0887131| 19.4362296| 0.0000000|
##
##
Food Density, Crime, Low Income
##
##
| | Estimate| Std. Error| t value| Pr(>|t|)|
|:------------|----------:|----------:|--------:|------------------:|
|(Intercept) | 368.098613| 18.9260189| 19.44934| 0|
|food_density | 9.656965| 0.8842676| 10.92086| 0|
|low_income | 593.696794| 27.6196207| 21.49547| 0|