```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)
library(here)
```

```
## here() starts at /Users/sid/Documents/ubc classes/2024w1/econ 326/foodprograms-326
```

```r
library(stargazer)
```

```
##
## Please cite as:
##
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

Census data

We load previously loaded census data. The code for fetching this data is also shown in this section.

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

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

```r
vectors <- c("v_CA21_1",
             "v_CA21_6",
             "v_CA21_452",
             "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"
)

census_data <- census_data %>%
  mutate(pop_density = `v_CA21_1: Population, 2021` / `Shape Area`)

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

```r
load(here("data/census.rda"))

n <- nrow(census_data)
head(census_data)
```

```
## Simple feature collection with 6 features and 22 fields
## Attribute-geometry relationships: constant (21), NA's (1)
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -123.0309 ymin: 49.27742 xmax: -123.0231 ymax: 49.29349
## Geodetic CRS:  WGS 84
##   Shape Area Type Households Quality Flags     name   GeoUID CSD_UID Population
## 1     0.2991   DA        266             0 59150307 59150307 5915022        677
## 2     0.1096   DA        218             0 59150308 59150308 5915022        541
## 3     0.1119   DA        282             0 59150309 59150309 5915022        761
## 4     0.1094   DA        389             0 59150310 59150310 5915022        748
## 5     0.0809   DA        187             0 59150311 59150311 5915022        537
## 6     0.0871   DA        201             0 59150312 59150312 5915022        555
##       CT_UID Dwellings CD_UID CMA_UID Region Name Area (sq km)
## 1 9330053.02       297   5915   59933    59150307       0.2991
## 2 9330053.02       229   5915   59933    59150308       0.1096
## 3 9330053.02       299   5915   59933    59150309       0.1119
## 4 9330053.02       411   5915   59933    59150310       0.1094
## 5 9330053.02       202   5915   59933    59150311       0.0809
## 6 9330053.02       218   5915   59933    59150312       0.0871
##   v_CA21_1: Population, 2021 v_CA21_6: Population density per square kilometre
## 1                        677                                            2263.5
## 2                        541                                            4936.1
## 3                        761                                            6800.7
## 4                        748                                            6837.3
## 5                        537                                            6637.8
## 6                        555                                            6372.0
##   v_CA21_452: Average household size
## 1                                2.6
## 2                                2.5
## 3                                2.5
## 4                                1.9
## 5                                2.9
## 6                                2.8
##   v_CA21_449: Number of persons in private households
## 1                                                 675
## 2                                                 540
## 3                                                 695
## 4                                                 750
## 5                                                 540
## 6                                                 555
##   v_CA21_1040: Prevalence of low income based on the Low-income measure, after tax (LIM-AT) (%)
## 1                                                                                           6.6
## 2                                                                                           7.2
## 3                                                                                           7.8
## 4                                                                                          13.2
## 5                                                                                          11.6
## 6                                                                                           8.0
##   v_CA21_1085: Prevalence of low income based on the Low-income cut-offs, after tax (LICO-AT) (%)
## 1                                                                                             5.2
## 2                                                                                             6.4
## 3                                                                                             5.8
## 4                                                                                             8.4
## 5                                                                                             9.6
## 6                                                                                             6.8
##   v_CA21_905: Income statistics for private households
## 1                                                  265
## 2                                                  220
## 3                                                  280
## 4                                                  390
## 5                                                  185
## 6                                                  200
##                         geometry pop_density
## 1 MULTIPOLYGON (((-123.0231 4...    2263.457
## 2 MULTIPOLYGON (((-123.0234 4...    4936.131
## 3 MULTIPOLYGON (((-123.0283 4...    6800.715
## 4 MULTIPOLYGON (((-123.0234 4...    6837.294
## 5 MULTIPOLYGON (((-123.0257 4...    6637.824
## 6 MULTIPOLYGON (((-123.0234 4...    6371.986
```

Food data

```
food_data <- st_read(here("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(here("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")
```

```r
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
```

```r
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
```

```r
kable(head(census_data_crime))
```

[illegible]									crimes_Vehicle Collision or Pedestrian Struck (with In-	[illegible]
[illegible] 45	0	2	0	0	6	0	1	0	3	MULTIPOLYGO (((-123.0231 4…

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	crimes_Vehicle Collision or Pedestrian Struck (with In-	[illegible]
[illegible]	6.4	[illegible]	3	0	2	7	0	1	0	0		MULTIPOLYGO (((-123.0234 4...
[illegible]	5.8	[illegible]	15	0	9	11	0	3	0	3		MULTIPOLYGO (((-123.0283 4...
[illegible]	[illegible]	[illegible]	16	0	19	7	3	3	0	5		MULTIPOLYGO (((-123.0234 4...
[illegible]	[illegible]	[illegible]	6	0	2	10	0	1	0	0		MULTIPOLYGO (((-123.0257 4...
[illegible]	6.8	[illegible]	3	0	0	7	0	2	0	2		MULTIPOLYGO (((-123.0234 4...

Training data

Now we merge all of our data-sets 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))
```

```r
training_data[is.na(training_data)] <- 0

kable(tail(training_data))
```

[illegible]									crimes_Vehicle Collision or Pedestrian Struck (with In-Shape	[illegible]	Quality	[illegible]
[illegible]	0	1	0	2	7	1	2	0	1	[illegible]	0	[illegible]
[illegible]	0	1	0	2	7	1	2	0	1	[illegible]	0	[illegible]
[illegible]	0	1	0	2	7	1	2	0	1	[illegible]	0	[illegible]
[illegible]	0	1	0	2	7	1	2	0	1	[illegible]	0	[illegible]
[illegible]	0	1	0	2	7	1	2	0	1	[illegible]	0	[illegible]
[illegible]	0	1	0	2	7	1	2	0	1	[illegible]	0	[illegible]

Model

Specification

```r
# Specification models
# 1. Model with all variables (including interaction term)
reg_all_vars <- lm(crime_density ~
                     food_density:low_income +
                     food_density +
                     pop_density.x +
                     v_CA21_452 +
                   low_income,
                 data = st_set_geometry(training_data, NULL))

# 2. Model without the interaction term
reg_no_interaction <- lm(crime_density ~
                         food_density +
                         pop_density.x +
                         v_CA21_452 +
                         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))

# 4. Model with all variables (including interaction term) but replacing v_CA21_452 with v_CA21_449
reg_all_vars_449 <- lm(crime_density ~
                       food_density:low_income +
                       food_density +
                       pop_density.x +
                       v_CA21_449 +  # Replaced v_CA21_452 with v_CA21_449
                       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)
summary_all_vars_449 <- summary(reg_all_vars_449)

# Add to the list of model summaries
model_summaries <- list(
  "All Variables" = summary_all_vars,
  "Without Interaction" = summary_no_interaction,
  "Food Density, Crime, Low Income" = summary_food_crime_low_income,
  "All Variables (with v_CA21_449)" = summary_all_vars_449
)
```

RESULTS FROM SPEC

```
stargazer(reg_all_vars, reg_no_interaction, reg_food_crime_low_income,
          type = "latex",
          covariate.labels = c("Food Program Density: Low Income",
                               "Food Program Density",
                               "Population Density",
                               "Average Household Size",
                               "Low Income"),
          dep.var.labels = "Crime Density",
          title = "Regression Models Summary",
          digits = 3)
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Sat, Nov 30, 2024 - 21:46:56

Table 4: Regression Models Summary

	Dependent variable:		
	Crime Density		
	(1)	(2)	(3)
Food Program Density: Low Income	−0.203	8.053***	9.631***
	(2.154)	(0.771)	(0.897)
Food Program Density	0.043***	0.044***	
	(0.001)	(0.001)	
Population Density	−517.756***	−517.272***	
	(22.439)	(22.461)	
Average Household Size	244.697***	252.689***	577.872***
	(24.734)	(24.682)	(27.744)
Low Income	9.460***		
	(2.305)		
Constant	1,291.286***	1,283.891***	383.969***
	(62.011)	(62.047)	(18.802)
Observations	7,860	7,860	7,860
R^2	0.315	0.313	0.067
Adjusted R^2	0.314	0.313	0.067
Residual Std. Error	1,050.245 (df = 7854)	1,051.304 (df = 7855)	1,225.029 (df = 7857)
F Statistic	720.960*** (df = 5; 7854)	895.186*** (df = 4; 7855)	282.621*** (df = 2; 7857)
Note:			*p<0.1; **p<0.05; ***p<0.01

```r
stargazer(reg_all_vars_449,
          type = "latex",
          covariate.labels = c("Food Program Density: Low Income",
                               "Food Program Density",
                               "Population Density",
                               "Persons in Household (v\\_CA21\\_449)",
                               "Low Income"),
          dep.var.labels = "Crime Density",
          title = "Regression Model Summary (with Persons in Household)",
          digits = 3)
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Sat, Nov 30, 2024 - 21:46:56

Table 5: Regression Model Summary (with Persons in Household)

	Dependent variable:
	Crime Density
Food Program Density: Low Income	1.513
	(2.218)
Food Program Density	0.053***
	(0.001)
Population Density	0.225***
	(0.031)
Persons in Household (v_CA21_449)	344.584***
	(25.039)
Low Income	9.063***
	(2.374)
Constant	−196.408***
	(25.778)
Observations	7,860
R^2	0.273
Adjusted R^2	0.273
Residual Std. Error	1,081.556 (df = 7854)
F Statistic	590.191*** (df = 5; 7854)
Note:	*p<0.1; **p<0.05; ***p<0.01