Census data

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

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

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

vectors <- c("v_CA21_1", "v_CA21_6", "v_CA21_452", "v_CA21_449", "v_CA21_1040", "v_CA21_1085", "v_CA21_9

region_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")

census_data <- get_census(
  dataset = "CA21",
  regions = list(CSD = "5915022", DA = region_DA),
  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")
```

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

n <- nrow(census_data)
kable(head(census_data[, 1:5]), format = "latex", booktabs = TRUE, caption = "Census Data, 2021 (truncat
  kable_styling(latex_options = c("striped", "hold_position"))
```

Table 1: Census Data, 2021 (truncated rows and columns)

Shape Area	Type	Households	Quality Flags	name	geometry
0.2991	DA	266	0	59150307	MULTIPOLYGON (((-123.0231 4...
0.1096	DA	218	0	59150308	MULTIPOLYGON (((-123.0234 4...
0.1119	DA	282	0	59150309	MULTIPOLYGON (((-123.0283 4...
0.1094	DA	389	0	59150310	MULTIPOLYGON (((-123.0234 4...
0.0809	DA	187	0	59150311	MULTIPOLYGON (((-123.0257 4...
0.0871	DA	201	0	59150312	MULTIPOLYGON (((-123.0234 4...

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[, 1:5]) %>%
  kable(format = "latex", booktabs = TRUE, caption = "Food Data merged with Census Data, 2021 (truncate
  kable_styling(latex_options = c("striped", "hold_position"))
```

—

Crime data

```
crime <- read_csv(here("data/crime_data_all_neighborhoods.csv"), show_col_types = FALSE) %>%
  mutate(TYPE = as_factor(TYPE),
         HUNDRED_BLOCK = as_factor(HUNDRED_BLOCK),
         NEIGHBOURHOOD = as_factor(NEIGHBOURHOOD)) %>%
```

Table 2: Food Data merged with Census Data, 2021 (truncated rows and columns)

Shape Area	Type	Households	Quality Flags	name	geometry
0.2991	DA	266	0	59150307	MULTIPOLYGON (((-123.0231 4...
0.1096	DA	218	0	59150308	MULTIPOLYGON (((-123.0234 4...
0.1119	DA	282	0	59150309	MULTIPOLYGON (((-123.0283 4...
0.1094	DA	389	0	59150310	MULTIPOLYGON (((-123.0234 4...
0.0809	DA	187	0	59150311	MULTIPOLYGON (((-123.0257 4...
0.0871	DA	201	0	59150312	MULTIPOLYGON (((-123.0234 4...

```
  filter(!is.na(X) & !is.na(Y))

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)

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
}
```

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))


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

Model

Specification

```
# 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 - 22:07:53

Table 3: 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

```
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 - 22:07:53

Table 4: 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