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_

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 (trunca
  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

```r
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/data/free-and-low-
cost-food-programs.shp'
##   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
```

```r
# 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 (truncated
  kable_styling(latex_options = c("striped", "hold_position"))
```

—

Crime data

```r
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),
```

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...

```
         NEIGHBOURHOOD = as_factor(NEIGHBOURHOOD)) %>%
  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), d
}
```

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))

models_spec <- list(
  "All Variables" = reg_all_vars,
  "Without Interaction" = reg_no_interaction,
  "Food Density, Crime, Low Income" = reg_food_crime_low_income
)

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

```r
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

Function definitions

We needed to manipulate our regression model data into formatted tables that emulated those from the package `stargazer` as we were having significant alignment issues between the data and rendered tables.

```r
all_variables <- unique(c(
  names(coef(reg_all_vars)),
  names(coef(reg_no_interaction)),
  names(coef(reg_food_crime_low_income))
))

all_variables_escaped <- gsub("_", "\\\\_", all_variables)  # Escape underscores
all_variables_escaped <- paste0("\\texttt{", all_variables_escaped, "}")  # Wrap in \texttt{} for LaTeX

add_stars <- function(estimates, ses, p_values) {
  significance_levels <- ifelse(p_values < 0.01, "***",
                          ifelse(p_values < 0.05, "**",
                          ifelse(p_values < 0.1, "*", "")))
  formatted <- sprintf("%.3f (%.3f)%s", estimates, ses, significance_levels)
  return(formatted)
}

extract_summary <- function(model, all_vars) {
  coefs <- coef(model)
  ses <- sqrt(diag(vcov(model)))
  p_values <- coef(summary(model))[, 4]  # Extract p-values from the summary

  # Create placeholders for all variables
  aligned_coefs <- setNames(rep(NA, length(all_vars)), all_vars)
  aligned_ses <- setNames(rep(NA, length(all_vars)), all_vars)
  aligned_p_values <- setNames(rep(NA, length(all_vars)), all_vars)

  # Fill with existing coefficients, standard errors, and p-values
  for (var in names(coefs)) {
    aligned_coefs[var] <- coefs[var]
    aligned_ses[var] <- ses[var]
    aligned_p_values[var] <- p_values[var]
  }

  # Replace missing coefficients with 0
  aligned_coefs[is.na(aligned_coefs)] <- 0
```

```r
  # Add stars to coefficients based on significance levels
  significance_levels <- ifelse(aligned_p_values < 0.01, "***",
                         ifelse(aligned_p_values < 0.05, "**",
                         ifelse(aligned_p_values < 0.1, "*", "")))

  # Format coefficients and standard errors for display
  formatted <- ifelse(is.na(aligned_ses),
                      sprintf("%.3f", aligned_coefs),
                      sprintf("%.3f (%.3f)%s", aligned_coefs, aligned_ses, significance_levels))

  # Replace missing standard errors with blank strings
  formatted[is.na(aligned_ses)] <- " "

  # Escape underscores for LaTeX compatibility and wrap in \texttt{} for table display
  formatted <- gsub("_", "\\\\_", formatted)
  formatted <- paste0("\\texttt{", formatted, "}")

  return(formatted)
}

extract_model_info <- function(model, covariate_labels, model_title) {
  coefs <- coef(model)
  ses <- sqrt(diag(vcov(model)))
  p_values <- coef(summary(model))[, 4]  # Extract p-values from the summary

  # Format coefficients, standard errors, and significance stars
  significance_levels <- ifelse(p_values < 0.01, "***",
                                ifelse(p_values < 0.05, "**",
                                       ifelse(p_values < 0.1, "*", "")))

  formatted <- sprintf("%.3f (%.3f)%s", coefs, ses, significance_levels)

  # Combine covariate labels and the formatted coefficients
  result <- data.frame(
    Variable = covariate_labels,
    Estimate = formatted,
    stringsAsFactors = FALSE
  )

  # Add model title
  result <- rbind(data.frame(Variable = model_title, Estimate = "", stringsAsFactors = FALSE), result)

  return(result)
}



# Extract R-squared, F-statistics, and number of observations from each model
r_squared <- c(
  summary(reg_all_vars)$r.squared,
  summary(reg_no_interaction)$r.squared,
  summary(reg_food_crime_low_income)$r.squared
)
```

```r
f_statistic <- c(
  summary(reg_all_vars)$fstatistic[1],
  summary(reg_no_interaction)$fstatistic[1],
  summary(reg_food_crime_low_income)$fstatistic[1]
)

n_obs <- c(
  length(reg_all_vars$fitted.values),
  length(reg_no_interaction$fitted.values),
  length(reg_food_crime_low_income$fitted.values)
)


model1_aligned <- extract_summary(reg_all_vars, all_variables)
model2_aligned <- extract_summary(reg_no_interaction, all_variables)
model3_aligned <- extract_summary(reg_food_crime_low_income, all_variables)
```

Data manipulation

```r
aligned_table <- data.frame(
  Variable = all_variables_escaped,
  `Model 1` = model1_aligned,
  `Model 2` = model2_aligned,
  `Model 3` = model3_aligned,
  stringsAsFactors = FALSE
)

print(aligned_table)
```

```
##                                Variable                          Model.1
## 1                 \\texttt{(Intercept)} \\texttt{1291.286 (62.011)***}
## 2              \\texttt{food\\_density}       \\texttt{-0.203 (2.154)}
## 3             \\texttt{pop\\_density.x}     \\texttt{0.043 (0.001)***}
## 4              \\texttt{v\\_CA21\\_452} \\texttt{-517.756 (22.439)***}
## 5               \\texttt{low\\_income}  \\texttt{244.697 (24.734)***}
## 6 \\texttt{food\\_density:low\\_income}     \\texttt{9.460 (2.305)***}
##                         Model.2                        Model.3
## 1 \\texttt{1283.891 (62.047)***} \\texttt{383.969 (18.802)***}
## 2     \\texttt{8.053 (0.771)***}    \\texttt{9.631 (0.897)***}
## 3     \\texttt{0.044 (0.001)***}                   \\texttt{ }
## 4 \\texttt{-517.272 (22.461)***}                   \\texttt{ }
## 5  \\texttt{252.689 (24.682)***} \\texttt{577.872 (27.744)***}
## 6                   \\texttt{ }                   \\texttt{ }
```

```r
summary_table <- data.frame(
  Variable = c("R squared", "F statistic", "Observations"),
  `Model 1` = c(sprintf("%.3f", r_squared[1]), sprintf("%.1f", f_statistic[1]), n_obs[1]),
  `Model 2` = c(sprintf("%.3f", r_squared[2]), sprintf("%.1f", f_statistic[2]), n_obs[2]),
  `Model 3` = c(sprintf("%.3f", r_squared[3]), sprintf("%.1f", f_statistic[3]), n_obs[3]),
  stringsAsFactors = FALSE
)

# Combine both tables
```

```r
final_table <- rbind(aligned_table, summary_table)
print(final_table)
```

```
##                               Variable                          Model.1
## 1                 \\texttt{(Intercept)} \\texttt{1291.286 (62.011)***}
## 2              \\texttt{food\\_density}       \\texttt{-0.203 (2.154)}
## 3             \\texttt{pop\\_density.x}     \\texttt{0.043 (0.001)***}
## 4              \\texttt{v\\_CA21\\_452} \\texttt{-517.756 (22.439)***}
## 5                \\texttt{low\\_income}  \\texttt{244.697 (24.734)***}
## 6 \\texttt{food\\_density:low\\_income}     \\texttt{9.460 (2.305)***}
## 7                            R squared                            0.315
## 8                          F statistic                            721.0
## 9                         Observations                             7860
##                         Model.2                      Model.3
## 1 \\texttt{1283.891 (62.047)***} \\texttt{383.969 (18.802)***}
## 2     \\texttt{8.053 (0.771)***}    \\texttt{9.631 (0.897)***}
## 3     \\texttt{0.044 (0.001)***}                   \\texttt{ }
## 4 \\texttt{-517.272 (22.461)***}                   \\texttt{ }
## 5  \\texttt{252.689 (24.682)***} \\texttt{577.872 (27.744)***}
## 6                   \\texttt{ }                   \\texttt{ }
## 7                         0.313                        0.067
## 8                         895.2                        282.6
## 9                          7860                         7860
```

```r
covariate_labels <- c(
  "(Intercept)",
  "Food Program Density: Low Income",
  "Food Program Density",
  "Population Density",
  "Average Household Size (v\\_CA21\\_452)",
  "Low Income"
)
covariate_labels_2 <- c(
  "(Intercept)",
  "Food Program Density",
  "Population Density",
  "Average Household Size (v\\_CA21\\_452)",
  "Low Income"
)

covariate_labels_3 <- c(
  "(Intercept)",
  "Food Program Density",
  "Low Income"
)

model_1_table <- extract_model_info(
  model = reg_all_vars,
  covariate_labels = covariate_labels,
  model_title = "Spec 1: All Variables"
)

model_2_table <- extract_model_info(
  model = reg_no_interaction,
```

```
  covariate_labels = covariate_labels_2,
  model_title = "Spec 2: Omitted Interaction Term"
)

model_3_table <- extract_model_info(
  model = reg_food_crime_low_income,
  covariate_labels = covariate_labels_3,
  model_title = "Spec 3: Food Density and Low Income"
)
```

Regression table

```
# Render the final table in a stargazer-like format using knitr::kable
kable(
  final_table,
  format = "latex",
  col.names = c("Variable", "Model 1", "Model 2", "Model 3"),
  caption = "Regression Specifications",
  align = "lccc",
  booktabs = TRUE,
  escape = FALSE
) %>%
  kable_styling(latex_options = c("hold_position", "striped")) %>%
  add_header_above(c(" " = 1, "Dependent Variable: Crime Density" = 3)) %>%
  footnote(general = "* p < 0.1; ** p < 0.05; *** p < 0.01",
           general_title = "Note:",
           footnote_as_chunk = TRUE,
           escape = FALSE)
```

Table 3: Regression Specifications

	Dependent Variable: Crime Density		
Variable	Model 1	Model 2	Model 3
(Intercept)	1291.286 (62.011)***	1283.891 (62.047)***	383.969 (18.802)***
food_density	-0.203 (2.154)	8.053 (0.771)***	9.631 (0.897)***
pop_density.x	0.043 (0.001)***	0.044 (0.001)***	
v_CA21_452	-517.756 (22.439)***	-517.272 (22.461)***	
low_income	244.697 (24.734)***	252.689 (24.682)***	577.872 (27.744)***
food_density:low_income	9.460 (2.305)***		
R squared	0.315	0.313	0.067
F statistic	721.0	895.2	282.6
Observations	7860	7860	7860

Note: * p < 0.1; ** p < 0.05; *** p < 0.01

```
stargazer(reg_all_vars, reg_no_interaction, reg_food_crime_low_income, type = "latex",
          covariate.labels = c("Food Program Density", "Population Density", "Household Size", "Low Inc
          dep.var.labels = c("Crime Density")
         )
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Tue, Dec 03, 2024 - 13:36:29

Table 4:

	Dependent variable:		
	Crime Density		
	(1)	(2)	(3)
Food Program Density	-0.203	8.053^{***}	9.631^{***}
	(2.154)	(0.771)	(0.897)
Population Density	0.043^{***}	0.044^{***}	
	(0.001)	(0.001)	
Household Size	-517.756^{***}	-517.272^{***}	
	(22.439)	(22.461)	
Low Income	244.697^{***}	252.689^{***}	577.872^{***}
	(24.734)	(24.682)	(27.744)
Food Density: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

Model 1:

```
# Model 1: reg_all_vars
stargazer(reg_all_vars, type = "latex",
          covariate.labels = c("Food Program Density", "Population Density", "Household Size", "Low Inco
          dep.var.labels = c("Crime Density")
         )
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Tue, Dec 03, 2024 - 13:36:29

Table 5:

	Dependent variable:
	Crime Density
Food Program Density	−0.203
	(2.154)
Population Density	0.043***
	(0.001)
Household Size	−517.756***
	(22.439)
Low Income	244.697***
	(24.734)
Food Density:Low Income	9.460***
	(2.305)
Constant	1,291.286***
	(62.011)
Observations	7,860
R^2	0.315
Adjusted R^2	0.314
Residual Std. Error	1,050.245 (df = 7854)
F Statistic	720.960*** (df = 5; 7854)
Note:	*p<0.1; **p<0.05; ***p<0.01

Model 2:

```
stargazer(reg_no_interaction, type = "latex",
          covariate.labels = c("Food Program Density", "Population Density", "Household Size", "Low Inco
          dep.var.labels = c("Crime Density")
         )
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Tue, Dec 03, 2024 - 13:36:29

Table 6:

	Dependent variable:
	Crime Density
Food Program Density	8.053***
	(0.771)
Population Density	0.044***
	(0.001)
Household Size	−517.272***
	(22.461)
Low Income	252.689***
	(24.682)
Constant	1,283.891***
	(62.047)
Observations	7,860
R^2	0.313
Adjusted R^2	0.313
Residual Std. Error	1,051.304 (df = 7855)
F Statistic	895.186*** (df = 4; 7855)
Note:	*p<0.1; **p<0.05; ***p<0.01

Model 3:

```
# Model 3: reg_food_crime_low_income
stargazer(reg_food_crime_low_income, type = "latex",
          covariate.labels = c("Food Program Density", "Low Income"),
          dep.var.labels = c("Crime Density")
         )
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Tue, Dec 03, 2024 - 13:36:29

Table 7:

	Dependent variable:
	Crime Density
Food Program Density	9.631***
	(0.897)
Low Income	577.872***
	(27.744)
Constant	383.969***
	(18.802)
Observations	7,860
R^2	0.067
Adjusted R^2	0.067
Residual Std. Error	1,225.029 (df = 7857)
F Statistic	282.621*** (df = 2; 7857)
Note:	*p<0.1; **p<0.05; ***p<0.01

Robustness

We split the data into sub-groups based on

```
census_data_centroids <- census_data %>%
  mutate(
    centroid = st_centroid(geometry),
    latitude = st_coordinates(centroid)[, 2],
    longitude = st_coordinates(centroid)[, 1]
  ) %>%
  st_set_geometry(NULL)  # Remove geometry for k-means input

# Normalize latitude and longitude for clustering
census_data_normalized <- census_data_centroids %>%
  mutate(
    latitude_scaled = scale(latitude),
    longitude_scaled = scale(longitude)
  )


set.seed(123) # For reproducibility
```

```r
k <- 4

kmeans_result <- kmeans(census_data_normalized[, c("latitude_scaled", "longitude_scaled")], centers = k

# Add cluster assignments to the original data
census_data$cluster <- kmeans_result$cluster

# Now, merge the cluster assignments into your training data
training_data_clusters <- st_join(training_data,
                         census_data %>% select(cluster, geometry),
                         left = TRUE)
```

```r
# Run the model for each cluster
cluster_models <- list()

for (i in 1:k) {
  subset_data <- training_data_clusters %>% filter(cluster == i)

  model <- lm(crime_density ~
              food_density:low_income +
              food_density +
              pop_density.x +
              v_CA21_452 +
              low_income,
            data = st_set_geometry(subset_data, NULL))

  cat(paste("\nModel Summary for Cluster", i, ":\n"))
  cluster_models[[i]] <- model
  print(summary(model)) # Check the results for each cluster
}
```

```
##
## Model Summary for Cluster 1 :
##
## Call:
## lm(formula = crime_density ~ food_density:low_income + food_density +
##     pop_density.x + v_CA21_452 + low_income, data = st_set_geometry(subset_data,
##     NULL))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1216.1  -147.9   -59.4    51.4  4377.5
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             8.200e+02  1.759e+01  46.611   <2e-16 ***
## food_density            6.652e-01  2.692e-01   2.471   0.0135 *
## pop_density.x           2.683e-02  5.950e-04  45.102   <2e-16 ***
## v_CA21_452             -2.569e+02  6.370e+00 -40.326   <2e-16 ***
## low_income             -1.282e+01  6.522e+00  -1.965   0.0494 *
## food_density:low_income -2.648e-01  3.629e-01  -0.730   0.4657
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 321.8 on 15892 degrees of freedom
## Multiple R-squared:  0.2516, Adjusted R-squared:  0.2514
## F-statistic:  1068 on 5 and 15892 DF,  p-value: < 2.2e-16
##
##
## Model Summary for Cluster 2 :
##
## Call:
## lm(formula = crime_density ~ food_density:low_income + food_density +
##     pop_density.x + v_CA21_452 + low_income, data = st_set_geometry(subset_data,
##     NULL))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4251.5 -1104.8  -406.5   361.3 10754.9
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.079e+03  9.577e+01  32.156  < 2e-16 ***
## food_density           -7.841e+00  5.525e+00  -1.419  0.15587
## pop_density.x           2.622e-02  1.052e-03  24.929  < 2e-16 ***
## v_CA21_452             -1.397e+03  5.016e+01 -27.842  < 2e-16 ***
## low_income              9.276e+02  3.169e+01  29.269  < 2e-16 ***
## food_density:low_income  1.671e+01  5.552e+00   3.010  0.00262 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1847 on 16445 degrees of freedom
## Multiple R-squared:  0.1932, Adjusted R-squared:  0.1929
## F-statistic: 787.4 on 5 and 16445 DF,  p-value: < 2.2e-16
##
##
## Model Summary for Cluster 3 :
##
## Call:
## lm(formula = crime_density ~ food_density:low_income + food_density +
##     pop_density.x + v_CA21_452 + low_income, data = st_set_geometry(subset_data,
##     NULL))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -272.32  -71.09  -22.73   34.85 1205.21
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             2.369e+02  7.746e+00  30.583  < 2e-16 ***
## food_density            9.935e-01  2.953e-01   3.364  0.00077 ***
## pop_density.x           1.591e-02  1.574e-04 101.094  < 2e-16 ***
## v_CA21_452             -5.109e+01  2.627e+00 -19.451  < 2e-16 ***
## low_income              5.528e+00  2.231e+00   2.478  0.01322 *
## food_density:low_income -2.823e+00  4.270e-01  -6.610 3.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 133.4 on 18348 degrees of freedom
## Multiple R-squared:  0.385,  Adjusted R-squared:  0.3849
## F-statistic:  2298 on 5 and 18348 DF,  p-value: < 2.2e-16
##
##
## Model Summary for Cluster 4 :
##
## Call:
## lm(formula = crime_density ~ food_density:low_income + food_density +
##     pop_density.x + v_CA21_452 + low_income, data = st_set_geometry(subset_data,
##     NULL))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -415.34  -62.84  -24.39   41.99  783.31
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.664e+01  4.931e+00  13.513  < 2e-16 ***
## food_density            5.133e+00  4.278e-01  11.997  < 2e-16 ***
## pop_density.x           1.237e-02  2.701e-04  45.811  < 2e-16 ***
## v_CA21_452             -1.375e+01  1.908e+00  -7.206 6.13e-13 ***
## low_income              3.670e+01  2.923e+00  12.556  < 2e-16 ***
## food_density:low_income -5.226e-01  6.081e-01  -0.859     0.39
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 127.7 on 11341 degrees of freedom
## Multiple R-squared:  0.2003, Adjusted R-squared:  0.1999
## F-statistic:   568 on 5 and 11341 DF,  p-value: < 2.2e-16
```

```r
stargazer(cluster_models, type = "latex",
          covariate.labels = c("Food Program Density", "Population Density", "Household Size", "Low Inco
          dep.var.labels = c("Crime Density"))
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Tue, Dec 03, 2024 - 13:36:30

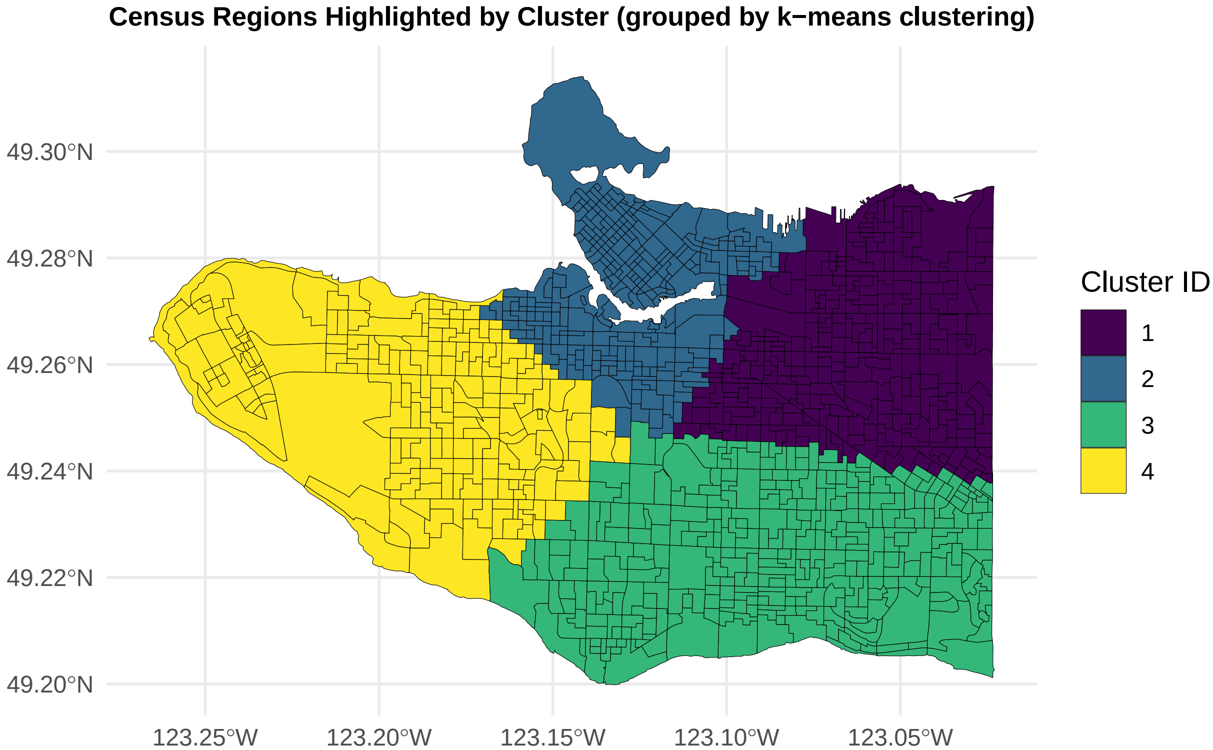
```r
# Convert the cluster column to a factor
census_data$cluster <- as.factor(census_data$cluster)

# Plot the regions and highlight clusters with adjusted color scale
ggplot(data = census_data) +
  geom_sf(aes(fill = cluster), color = "black", lwd = 0.1) +
  scale_fill_viridis_d(name = "Cluster ID") +  # Automatically assign colors for clusters
  labs(title = "Census Regions Highlighted by Cluster (grouped by k-means clustering)",
       fill = "Cluster ID") +
  theme_minimal() +
  theme(legend.position = "right",
        plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5, size = 12))
```

Table 8:

	Dependent variable:		
	Crime Density		
	(1)	(2)	(3)
Food Program Density	0.665**	−7.841	0.994***
	(0.269)	(5.525)	(0.295)
Population Density	0.027***	0.026***	0.016***
	(0.001)	(0.001)	(0.0002)
Household Size	−256.874***	−1,396.640***	−51.092***
	(6.370)	(50.163)	(2.627)
Low Income	−12.819**	927.569***	5.528**
	(6.522)	(31.691)	(2.231)
Food Density:Low Income	−0.265	16.712***	−2.823***
	(0.363)	(5.552)	(0.427)
Constant	819.960***	3,079.475***	236.881***
	(17.592)	(95.768)	(7.746)
Observations	15,898	16,451	18,354
R^2	0.252	0.193	0.385
Adjusted R^2	0.251	0.193	0.385
Residual Std. Error	321.788 (df = 15892)	1,846.752 (df = 16445)	133.450 (df = 18348)
F Statistic	1,068.458*** (df = 5; 15892)	787.362*** (df = 5; 16445)	2,297.625*** (df = 5; 18348)

Note: *p

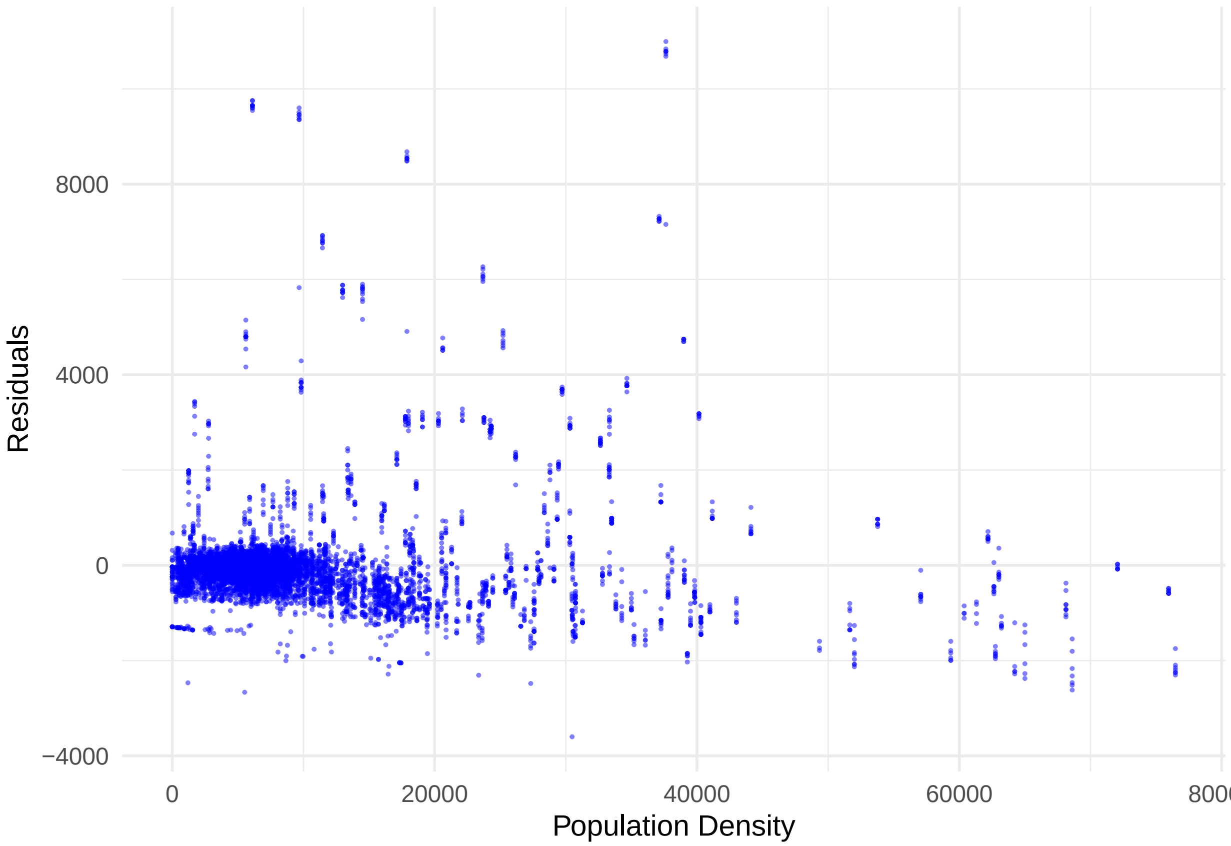


Heteroskedascity

```
# Assuming you have a model named 'reg_all_vars' and a dataset named 'training_data'

# Add residuals to the dataset
training_data_res <- training_data %>%
  mutate(residuals = resid(reg_all_vars))

# Plot residuals against a variable (e.g., population density) to visually check for heteroskedasticity
ggplot(data = training_data_res, aes(x = pop_density.x, y = residuals)) +
  geom_point(size = 0.2, alpha =0.5, color="blue") +
  labs(x = "Population Density", y = "Residuals") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Extract residuals and calculate squared residuals
training_data_res <- training_data_res %>%
  mutate(residuals_squared = residuals^2)

# Auxiliary regression: squared residuals on independent variables, their squares, and interactions
white_model <- lm(residuals_squared ~
                    food_density + pop_density.x + v_CA21_452 + low_income +
                    I(food_density^2) + I(pop_density.x^2) + I(v_CA21_452^2) + I(low_income^2) +
                    food_density:pop_density.x + food_density:v_CA21_452 + food_density:low_income +
                    pop_density.x:v_CA21_452 + pop_density.x:low_income +
                    v_CA21_452:low_income,
                  data = training_data_res)

# Calculate the test statistic
n <- nrow(training_data)
R_squared <- summary(white_model)$r.squared
white_test_statistic <- n * R_squared

# Calculate p-value for the test statistic (chi-square with degrees of freedom equal to number of predi
p_value <- pchisq(white_test_statistic, df = length(coef(white_model)) - 1, lower.tail = FALSE)

cat("White's Test Statistic:", white_test_statistic, "\n")
```

```
## White's Test Statistic: 644.1092
```

```
cat("p-value:", p_value, "\n")
```

```
## p-value: 2.147077e-128
```

```r
# Auxiliary regression for Breusch-Pagan Test: regress squared residuals on original predictors
bp_model <- lm(residuals_squared ~
                 food_density + pop_density.x + v_CA21_452 + low_income,
               data = training_data_res, na.action = na.omit)

# Calculate the test statistic for Breusch-Pagan test
R_squared_bp <- summary(bp_model)$r.squared
bp_test_statistic <- n * R_squared_bp

# Calculate p-value for Breusch-Pagan test statistic (chi-square)
p_value_bp <- pchisq(bp_test_statistic, df = length(coef(bp_model)) - 1, lower.tail = FALSE)

cat("Breusch-Pagan Test Statistic:", bp_test_statistic, "\n")
```

```
## Breusch-Pagan Test Statistic: 436.9128
```

```r
cat("Breusch-Pagan Test p-value:", p_value_bp, "\n")
```

```
## Breusch-Pagan Test p-value: 2.930512e-93
```

```r
# Calculate robust standard errors and re-run the model with robust covariance matrix
robust_se <- coeftest(reg_all_vars, vcov = vcovHC(reg_all_vars, type = "HC"))

# Print the results with robust standard errors

stargazer(robust_se, type = "latex",
          covariate.labels = c("Food Program Density", "Population Density", "Household Size", "Low Inc
          dep.var.labels = c("Crime Density")
         )
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Tue, Dec 03, 2024 - 13:36:31

Table 9:

	Dependent variable:
	Crime Density
Food Program Density	−0.203
	(0.545)
Population Density	0.043***
	(0.002)
Household Size	−517.756***
	(28.164)
Low Income	244.697***
	(21.468)
Food Density:Low Income	9.460***
	(2.844)
Constant	1,291.286***
	(79.646)
Note:	*p<0.1; **p<0.05; ***p<0.01