Data Analysis Final Project

Hariharasudhan Giridharan, Dharmasurya Arulmozhi, Allotei Pappoe

22/3/2024

A Abstract

The adoption of EVs is a key part in the effort of nations such as Canada to address climate change. In order to ensure the growth of this technology continues in an optimal manner, it is necessary to understand where that growth will take place. Adoption of electric vehicles is not expected to be evenly distributed across the population, or across geographic areas. Different people have different transportation needs and capabilities, and this will determine whether they will consider the use of an electric vehicle. Factors such as income, employment field, housing, education, commuting patterns, personal characteristics and access to charging infrastructure may all affect this choice. We seek to understand how these factors affect the adoption of EVs. Our ultimate goal is to compare different models that perform well on our dataset and eventually use the census profile data from different provinces to predict the total presence of electric vehicles. We are aiming to train our data from the Ontario's census data initially.

B Keywords

Generalized Linear models, Electric Vehicle Adoption, Forward Sortation Area (FSA), Random Forests.

C Introduction

The Electric Vehicle (EV) market in Canada is booming. By September 2023, out of 1,286,951 registered vehicles, 132,783 were electric. This growth aligns with Canada's climate goals, aiming to phase out gas vehicles by 2035 and have EVs make up 20% of sales by 2026. Regulations are driving investment in EV infrastructure. From 2017 to 2021, EV registrations surged from under 20,000 to over 86,000, with 7.7% of all vehicles registered as electric by early 2022. Consumer interest is high, with 71% considering an EV for their next purchase, demanding a minimum range of 400 kilometers per charge (Cousin, 2023; Blair, 2024).

The transition towards electric vehicles (EVs) presents a crucial step in addressing environmental concerns and promoting sustainable transportation. Organizations and businesses are increasingly interested in understanding and predicting EV adoption patterns. This interest stems from the need to develop effective policies and strategies that encourage EV usage and to anticipate the infrastructural demands associated with a growing EV market. Recent studies have shown that various demographic, technical, economic, and behavioral factors significantly influence EV adoption. For instance, (Egbue et al., 2017) highlighted the importance of demographic determinants and behavioral attitudes in influencing individual adoption decisions (Egbue et al., 2017). Similarly, (Chen et al., 2020) emphasized the interconnected influence of socio-demographics and behavioral factors on EV adoption interest (Chen et al., 2020).

D Data Description

To make predictions regarding uptake of electric vehicles in different regions within Ontario on the basis of population characteristics, we require data regarding use and/or ownership of EVs, divided by region, and a variety of demographic figures for these same regions. The Ontario Ministry of Transportation makes data available regarding the number of registered EVs in the province divided by Forward Sortation Area (FSA), i.e., the first three digits in an area's postal code. This dataset contains only four columns.

Variable Name	Type	Description
FSA	Categorical/Identifier (text)	Identifies the individual areas (instances)
PHEV	Continuous (numeric)	Hybrid EVs registered in the area
BEV	Continuous (numeric)	Battery-powered EVs registered in the area
TotalEV	Continuous (numeric)	Total EVs registered in the area

Table 1: Variables in Electric Vehicles in Ontario - By Forward Sortation Area dataset

The FSA column identifies the individual areas for which we will make our predictions. The TotalEV column will serve as the basis for our target variable; however, this is the raw number of registrations in an area, which may depend on population size. This dataset contains 550 instances, each representing a distinct geographic area. Some registrations may be tied to FSAs that represent areas without a residential population, and will be removed from consideration.

Statistics Canada provides detailed information on many aspects of the Canadian populace, including the domain concepts listed above. We downloaded the CSV-formatted version of the 2021 Census Profile data, organized by FSA, corresponding to the areas listed in the EV dataset. This dataset contains data on all 1646 FSAs in Canada, including 521 within Ontario.

Each row in this dataset corresponds to a "characteristic" of an area, with columns representing metadata, total counts, and breakdowns by gender. The CSV contains 23 columns, however only five are of any interest to us.

Variable Name	Type	Description
GEO_NAME	Categorical/Identifier (text)	Equivalent to FSA
CHARACTERISTIC_ID	–	Identifies the variable/characteristic
CHARACTERISTIC_NAME	–	Identifies the variable/characteristic
C1_COUNT_TOTAL	Continuous (numeric)	Provides the raw data for most features
C2_COUNT_MEN+	Continuous (numeric)	Provides the raw data for the "num_males" feature

Table 2: Variables in Dataset

There are 2,631 characteristics (rows) for each FSA, resulting in 1,370,751 total rows of interest. We selected 143 characteristics, corresponding to the domains we identified as possibly yielding features from which EV ownership/registration may be predicted, as well as those necessary to normalize other values. After selecting these, the table was transposed to a format with one row per FSA, and each column representing a selected feature.

The EV and Census Profile datasets were joined on the FSA identifiers, resulting in 517 observations (FSAs present in both data sets) with 147 columns. Four FSAs with very small or no populations were removed. Statistics referring to counts of individual people were divided by the overall population of the area, while those referring to counts of dwellings (homes) were divided by the number of dwellings in the area. The number of registered EVs were divided by the population and multiplied by 10,000 to yield the number of registered EVs per 10,000 people.

As the TotalEV number represents only the number of registered vehicles at a particular moment in time, and it is generally expected not only that this number will increase, but that the overall adoption rate will

increase, rather than attempting to predict this number via regression, we decided to use it as the basis for a new categorical variable, TotalEV_Category. This variable will divide our dataset into areas of Low, Medium, or High EV adoption, based on analysis below.

After assembling our data and normalizing it over the population, we ran a simple correlation check between the possible features and the (normalized) TotalEV value. We selected 26 features from across multiple domains with the highest correlations (positive or negative) in their categories. Note that only "med_fulltime_income" would qualify as being particularly high, but other features may still contribute in some ways.

The source data is relatively high-quality, and requires little in the way of cleaning, beyond the normalization we have already done. Missing values only appear to exist for those regions with very small populations; we have removed four FSAs from consideration as a result.

The data scheme is represented as follows:

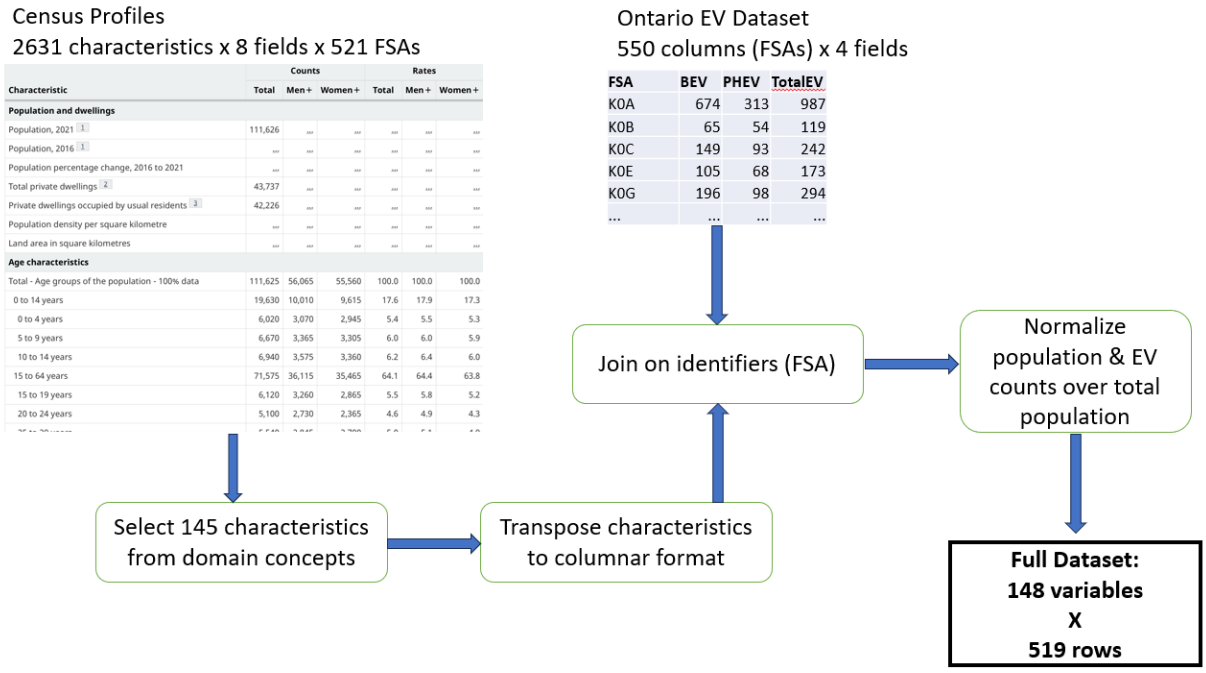


Figure 1: Data Flow pipeline

The final dataset after binning similar features from correaltion matrix is as follows:

Research Questions

RQ1. How does the demographics of a Forward Sortation Area affect the TotalEV's count?

RQ2. Across Ontario, what is the effect of the count of charging stations for electric vehicles on the count of electric vehicles? Are there any FSAs which are hotspots or coldspots for electric vehicles?

RQ3. Can tree based methods be used to predict TotalEvs based on demographical variables? Per tree based models, what demographical variables are the most important in predicting TotalEVs?

Report Organization

Variable	Explanation
med_fulltime_income	Median income of full-time employees in the FSA
income_100k_up	Number of people with income over $100k in the FSA
med_total_household_income	Median total household income in the FSA
income_40k	Number of people with income over $40k in the FSA
occup_cat_snr_mgmt	Number of people in senior management occupations in the FSA
occup_cat_busfin	Number of people in business and finance occupations in the FSA
occup_ind_realestate	Number of people in the real estate industry in the FSA
work_loc_home	Number of people who work from home in the FSA
work_loc_workplace	Number of people who work at a workplace (not from home) in the FSA
work_loc_notfixed	Number of people with non-fixed work locations in the FSA
school_uni_degree	Number of people with a university degree in the FSA
school_hs	Number of people with a high school diploma in the FSA
commute_start_noon	Number of people who start their commute at noon in the FSA
edu_field_science	Number of people with education in science fields in the FSA
edu_field_bus_admin	Number of people with education in business administration in the FSA
edu_field_health	Number of people with education in health fields in the FSA
edu_field_pers_protect_transp	Number of people with education in personal protection or transportation in the FSA
commute_transp_carpass	Number of people who commute as car passengers in the FSA
commute_start_6am	Number of people who start their commute at 6 AM in the FSA
commute_start_9am	Number of people who start their commute at 9 AM in the FSA
condo	Number of condominium units in the FSA
worktype_selfemp	Number of self-employed individuals in the FSA
indigenous_ppl	Number of indigenous people in the FSA
married_ppl	Number of married people in the FSA
work_loc_foreign	Number of people who work at foreign locations in the FSA
can_citizen_ppl	Number of Canadian citizens in the FSA
non_citizen_ppl	Number of non-citizens in the FSA

Table 3: Explanation of Variables in the Forward Sortation Area (FSA)

The subsequent sections of this report are organized as the following sections: E) A methods section that will describe exploratory data analysis, briefly outline models used that were already covered in class material, and review in more detail approaches specific to this project or not discussed extensively in class. F) An analysis section consisting of R code for producing the visualization and model fits outlined in Methods with concise commenting. G) A Results section that serves as a results and discussion of the main finding and their warranted interpretations relevant to informative insights about the research questions.

E Methods

We carried out exploratory data analysis (EDA) that included scatter plots for all variables and visualizations of response variables across various combinations of potentially confounding explanatory effects, which we will refer to as influence scenarios. The aim of this EDA was to identify the distributions of responses and the correlation structures present in the data, which could be categorized by significant scenarios. These results informed our decisions regarding the selection of appropriate models. Additionally, we examined summary statistics for the explanatory variables, paying special attention to missing values.

To address the first research question, we began by performing data normalization as previously described. We plotted a histogram to determine if the response variable exhibited a normal distribution. Based on the results obtained, we attempted to fit a Generalized Linear Model (GLM) to our dataset and further expand our research based on it. Considering the skewness of the response variable, we initially fitted the model

with a Poisson distribution and then built upon it using quasi-Poisson and negative binomial distributions. We plotted the model's residual plots in three different ways and concluded that the negative binomial distribution provided the best fit among the three distributions, a realization also supported by the Akaike Information Criterion (AIC). Subsequently, we utilized the Incidence Rate Ratio (IRR) to identify the top five demographic features that play a significant role in the total electric vehicle count.

For the second research question, our study aims to assess the effect of the availability of electric vehicle (EV) charging stations on the adoption of electric vehicles across Ontario. Furthermore, it seeks to identify spatial hotspots and coldspots for EVs within the province. With the aforementioned datasets, for this question we wrangled and combined the count of EVs and EV charging stations for each forward sortation area (FSA) in Ontario ensuring a consistent spatial framework for analysis. This resulted in a dataset featuring FSAs, counts of total EVs and counts of EV charging stations.
We conducted EDA to understand the distribution, mean and variance of the counts of EV and charging stations across Ontario. Spatial distributions of both EVs and charging stations were then visualised using maps of Ontario. This was done to help in identifying preliminary patterns, such as areas with high concentrations of EVs and or charging stations.
For statistical modelling, we examined the distribution of the counts of EVs to determine the most appropriate model to use. Based on the distribution identified, a GLM with Poisson family of distribution was initially fitted to analyse the relationship between the count of EV charging stations and the count of EVs. After careful analysis of the data and running overdispersion tests, we realized the variance is far greater than the mean of the data. A Negative Binomial GLM proved to be a better fit for the data. We then analysed the diagnostic plots and performed residual checking to ensure the chosen model was a right fit. We then interpreted the coefficients of the fitted model to understand the effect of EV charging stations on the adoption of electric vehicles.
For the hotspot analysis, we used spatial statistical methods to identify FSAs that are hotspots and coldspots (areas with significantly high counts of EVs).

For our third question, we aimed to evaluate the efficacy of tree-based models (including Decision Trees, Random Forests, and Gradient Boosting Machines) in predicting TotalEVs from demographic variables and to identify which demographic variables are the most influential in predicting the presence of TotalEVs, thereby providing insights into factors driving EV adoption. The methodology for assessing the potential of tree-based methods to predict TotalEVs based on demographic variables incorporates a comprehensive approach that leverages Decision Trees, Random Forests, and Gradient Boosting Machines (XGBoost). These models are chosen for their ability to handle complex, nonlinear relationships between a large number of predictor variables and the outcome, making them ideal for this research question.

F Analysis

The code described has been utilized for the necessary data normalization process. We have employed the Census dataset to examine the demographics, select specific features, and reformat these features into a columnar structure. Specifically, we selected the Census dataset from the Stats Canada Ontario website to analyze the total count of electric vehicles (EVs) based on the demographics provided in the Census dataset. The objective of this project is to amalgamate these two datasets by linking them through the Forward Sortation Area (FSA) and integrating their data. To achieve this, it was essential to extract the relevant features from the extensive dataset. Although the dataset contains a vast array of features, we determined that many were superfluous and thus, only included the necessary features. The approach to normalization we adopted is somewhat intuitive, aiming to mitigate the bias attributed to population size in the total count of electric vehicles (the dependent variable). It's evident that a higher population in a specific FSA is likely to correspond to a higher number of EVs in that area. Therefore, we normalized the features (excluding income features) by dividing the population count for each feature by the total population in that FSA, per 10,000 individuals.

```
# Re-organize census data

# library(tidyverse)
#
# setwd("~/Documents/School/CSCI6409/Project")
#
# data = read.csv("Data/Census\ Profile/98-401-X2021013_English_CSV_data.csv")
#
# data <- data %>%
#   filter(substr(GEO_NAME, 1, 1) %in% list('K','L','M','N','P')) %>%
#   select(-CENSUS_YEAR, -DGUID, -ALT_GEO_CODE, -GEO_LEVEL, -TNR_SF, -TNR_LF, -DATA_QUALITY_FLAG, -CHAR
#          -SYMBOL, -SYMBOL.1, -SYMBOL.2, -SYMBOL.3, -SYMBOL.4, -SYMBOL.5) %>%
#   filter(CHARACTERISTIC_ID %in% list(1,4,6,7,8,16,17,18,19,20,21,22,23,24,39,40,
#                                      42,43,44,45,46,47,48,49,57,59,66,128,130,143,144,
#                                      156,158,159,160,161,162,163,164,165,166,167,168,
#                                      243,244,1403,1410,1415,1416,1417,1419,1420,1433,
#                                      1466,1467,1523,1526,1528,1529,1537,1975,1976,1984,1985,
#                                      2015,2016,2018,2024,2095,2097,2100,2109,2117,2121,2127,2132,2140
#                                      2149,2155,2228,2229,2230,2232,2233,2234,2235,2236,2240,2245,
#                                      2249,2250,2251,2252,2253,2254,2255,2256,2257,2258,
#                                      2262,2263,2264,2265,2266,2267,2268,2269,2270,
#                                      2271,2272,2273,2274,2275,2276,2277,2278,2279,2280,2281,
#                                      2594,2595,2596,2597,
#                                      2599,2600,2601,2602,
#                                      2605,2606,2607,2608,2609,2610,
#                                      2612,2613,2614,2615,2616,
#                                      2618,2619,2620,2621,2622,2623)) %>%
#   mutate(value = ifelse(CHARACTERISTIC_ID==8, C2_COUNT_MEN., C1_COUNT_TOTAL)) %>%
#   select(GEO_NAME, CHARACTERISTIC_ID, CHARACTERISTIC_NAME, value)
#
# field_names = tibble(
#   CHARACTERISTIC_ID = c(1,4,6,7,8,16,17,18,19,20,21,22,23,24,39,40,
#                         42,43,44,45,46,47,48,49,57,59,66,128,130,143,144,
#                         156,158,159,160,161,162,163,164,165,166,167,168,
#                         243,244,1403,1410,1415,1416,1417,1419,1420,1433,
#                         1466,1467,1523,1526,1528,1529,1537,1975,1976,1984,1985,
#                         2015,2016,2018,2024,2095,2097,2100,2109,2117,2121,2127,2132,2140,2143,
#                         2149,2155,2228,2229,2230,2232,2233,2234,2235,2236,2240,2245,
#                         2249,2250,2251,2252,2253,2254,2255,2256,2257,2258,
#                         2262,2263,2264,2265,2266,2267,2268,2269,2270,
#                         2271,2272,2273,2274,2275,2276,2277,2278,2279,2280,2281,
#                         2594,2595,2596,2597,
#                         2599,2600,2601,2602,
#                         2605,2606,2607,2608,2609,2610,
#                         2612,2613,2614,2615,2616,
#                         2618,2619,2620,2621,2622,2623),
#   char_col_name = c("total_pop","total_dwellings","pop_density","land_area","num_male",
#                     "age_25","age_30","age_35","age_40","age_45","age_50","age_55","age_60","age_6
#                     "homes_detached_house","homes_semidetached_house","homes_rowhouse","homes_dupl
#                     "homes_highrise_apt","homes_other_stationary","homes_mobile",
#                     "avg_ppl_household","married_ppl","single_ppl",
#                     "avg_total_income","avg_aftertax_income","med_fulltime_income","avg_fulltime_i
#                     "income_none","income_under_10k","income_10k","income_20k","income_30k","incom
```

```
#                         "income_70k","income_80k","income_90k","income_100k_up",
#                         "med_total_household_income","med_aftertax_household_income",
#                         "indigenous_ppl","nonindigenous_ppl",
#                         "home_owner","home_renter","home_gov_or_ind_band",
#                         "condo","non_condo","avg_rooms_home",
#                         "home_cost_under_30pct","home_cost_over_30pct",
#                         "can_citizen_ppl","non_citizen_ppl",
#                         "non_immigrant_ppl","immigrant_ppl","non_perm_res_ppl",
#                         "no_move_last_yr","moved_last_yr","no_move_5yrs","moved_last_5yrs",
#                         "school_no_hs","school_hs","school_college","school_uni_degree",
#                         "edu_field_education","edu_field_arts_comms","edu_field_humanities","edu_field
#                         "edu_field_science","edu_field_math_cs","edu_field_engin_arch","edu_field_agri
#                         "edu_field_pers_protect_transp","edu_field_other",
#                         "workforce_participation_rate","workforce_employment_rate","workforce_unemploy
#                         "work_lastyear_didnotwork","work_lastyear_worked","work_lastyear_fulltime","wo
#                         "worktype_employee","worktype_selfemp",
#                         "occup_cat_snr_mgmt","occup_cat_busfin","occup_cat_science","occup_cat_health"
#                         "occup_cat_sales_serv","occup_cat_trades_transp","occup_cat_natres_agr","occup
#                         "occup_ind_agr_forest","occup_ind_mine_og","occup_ind_util","occup_ind_constr"
#                         "occup_ind_retail_trd","occup_ind_transp_warehs","occup_ind_info_culture","occ
#                         "occup_ind_prof_sci_tech_serv","occup_ind_mgmt","occup_ind_admsupport_wastemgm
#                         "occup_ind_arts_ent_rec","occup_ind_accom_food_svc","occup_ind_other","occup_i
#                         "work_loc_home","work_loc_foreign","work_loc_notfixed","work_loc_workplace",
#                         "commute_same_subdiv","commute_same_div","commute_same_prov","commute_diff_pro
#                         "commute_transp_cardriver","commute_transp_carpass","commute_transp_pubtrans",
#                         "commute_transp_bike","commute_transp_other",
#                         "commute_time_under15","commute_time_15","commute_time_30","commute_time_45","
#                         "commute_start_5am","commute_start_6am","commute_start_7am","commute_start_8am
# )
#
# new_data <- inner_join(data, field_names, by='CHARACTERISTIC_ID') %>%
#   select(GEO_NAME, char_col_name, value) %>%
#   pivot_wider(id_cols = GEO_NAME, names_from = char_col_name, values_from = value)
#
#
# ev_data = read.csv("Data/ontario_evs_by_fsa_2023-03-31.csv")
#
# new_data <- inner_join(new_data, ev_data, by=join_by(GEO_NAME==FSA))
#
# new_data <- mutate(new_data,
#   num_male = num_male/total_pop,
#
#   age_25 = age_25/total_pop,
#   age_30 = age_30/total_pop,
#   age_35 = age_35/total_pop,
#   age_40 = age_40/total_pop,
#   age_45 = age_45/total_pop,
#   age_50 = age_50/total_pop,
#   age_55 = age_55/total_pop,
#   age_60 = age_60/total_pop,
#   age_65_up = age_65_up/total_pop,
#
#   homes_detached_house = homes_detached_house/total_dwellings,
```

```
#   homes_semidetached_house = homes_semidetached_house/total_dwellings,
#   homes_rowhouse = homes_rowhouse/total_dwellings,
#   homes_duplex_apt = homes_duplex_apt/total_dwellings,
#   homes_lowrise_apt = homes_lowrise_apt/total_dwellings,
#
#   homes_highrise_apt = homes_highrise_apt/total_dwellings,
#   homes_other_stationary = homes_other_stationary/total_dwellings,
#   homes_mobile = homes_mobile/total_dwellings,
#
#   married_ppl = married_ppl/total_pop,
#   single_ppl = single_ppl/total_pop,
#
#   income_none = income_none/total_pop,
#   income_under_10k = income_under_10k/total_pop,
#   income_10k = income_10k/total_pop,
#   income_20k = income_20k/total_pop,
#   income_30k = income_30k/total_pop,
#   income_40k = income_40k/total_pop,
#   income_50k = income_50k/total_pop,
#   income_60k = income_60k/total_pop,
#   income_70k = income_70k/total_pop,
#   income_80k = income_80k/total_pop,
#   income_90k = income_90k/total_pop,
#   income_100k_up = income_100k_up/total_pop,
#
#   indigenous_ppl = indigenous_ppl/total_pop,
#   nonindigenous_ppl = nonindigenous_ppl/total_pop,
#
#   home_owner = home_owner/total_pop,
#   home_renter = home_renter/total_pop,
#   home_gov_or_ind_band = home_gov_or_ind_band/total_pop,
#
#   condo = condo/total_dwellings,
#   non_condo = non_condo/total_dwellings,
#
#   home_cost_under_30pct = home_cost_under_30pct/total_pop,
#   home_cost_over_30pct = home_cost_over_30pct/total_pop,
#
#   can_citizen_ppl = can_citizen_ppl/total_pop,
#   non_citizen_ppl = non_citizen_ppl/total_pop,
#   non_immigrant_ppl = non_immigrant_ppl/total_pop,
#   immigrant_ppl = immigrant_ppl/total_pop,
#   non_perm_res_ppl = non_perm_res_ppl/total_pop,
#
#   no_move_last_yr = no_move_last_yr/total_pop,
#   moved_last_yr = moved_last_yr/total_pop,
#   no_move_5yrs = no_move_5yrs/total_pop,
#   moved_last_5yrs = moved_last_5yrs/total_pop,
#
#   school_no_hs = school_no_hs/total_pop,
#   school_hs = school_hs/total_pop,
#   school_college = school_college/total_pop,
#   school_uni_degree = school_uni_degree/total_pop,
```

```
#
#   edu_field_education = edu_field_education/total_pop,
#   edu_field_arts_comms = edu_field_arts_comms/total_pop,
#   edu_field_humanities = edu_field_humanities/total_pop,
#   edu_field_socsci_law = edu_field_socsci_law/total_pop,
#   edu_field_bus_admin = edu_field_bus_admin/total_pop,
#   edu_field_science = edu_field_science/total_pop,
#   edu_field_math_cs = edu_field_math_cs/total_pop,
#   edu_field_engin_arch = edu_field_engin_arch/total_pop,
#   edu_field_agri_res_env = edu_field_agri_res_env/total_pop,
#   edu_field_health = edu_field_health/total_pop,
#   edu_field_pers_protect_transp = edu_field_pers_protect_transp/total_pop,
#   edu_field_other = edu_field_other/total_pop,
#
#   work_lastyear_didnotwork = work_lastyear_didnotwork/total_pop,
#   work_lastyear_worked = work_lastyear_worked/total_pop,
#   work_lastyear_fulltime = work_lastyear_fulltime/total_pop,
#   work_lastyear_parttime = work_lastyear_parttime/total_pop,
#   work_lastyear_avg_weeks = work_lastyear_avg_weeks/total_pop,
#
#   worktype_employee = worktype_employee/total_pop,
#   worktype_selfemp = worktype_selfemp/total_pop,
#
#   occup_cat_snr_mgmt = occup_cat_snr_mgmt/total_pop,
#   occup_cat_busfin = occup_cat_busfin/total_pop,
#   occup_cat_science = occup_cat_science/total_pop,
#   occup_cat_health = occup_cat_health/total_pop,
#   occup_cat_edu_law_socserv = occup_cat_edu_law_socserv/total_pop,
#   occup_cat_arts_rec = occup_cat_arts_rec/total_pop,
#   occup_cat_sales_serv = occup_cat_sales_serv/total_pop,
#   occup_cat_trades_transp = occup_cat_trades_transp/total_pop,
#   occup_cat_natres_agr = occup_cat_natres_agr/total_pop,
#   occup_cat_manuf_util = occup_cat_manuf_util/total_pop,
#
#   occup_ind_agr_forest = occup_ind_agr_forest/total_pop,
#   occup_ind_mine_og = occup_ind_mine_og/total_pop,
#   occup_ind_util = occup_ind_util/total_pop,
#   occup_ind_constr = occup_ind_constr/total_pop,
#   occup_ind_manuf = occup_ind_manuf/total_pop,
#   occup_ind_wholesale_trd = occup_ind_wholesale_trd/total_pop,
#   occup_ind_retail_trd = occup_ind_retail_trd/total_pop,
#   occup_ind_transp_warehs = occup_ind_transp_warehs/total_pop,
#   occup_ind_info_culture = occup_ind_info_culture/total_pop,
#   occup_ind_fin_insure = occup_ind_fin_insure/total_pop,
#   occup_ind_realestate = occup_ind_realestate/total_pop,
#   occup_ind_prof_sci_tech_serv = occup_ind_prof_sci_tech_serv/total_pop,
#   occup_ind_mgmt = occup_ind_mgmt/total_pop,
#   occup_ind_admsupport_wastemgmt = occup_ind_admsupport_wastemgmt/total_pop,
#   occup_ind_edu = occup_ind_edu/total_pop,
#   occup_ind_health_socasst = occup_ind_health_socasst/total_pop,
#   occup_ind_arts_ent_rec = occup_ind_arts_ent_rec/total_pop,
#   occup_ind_accom_food_svc = occup_ind_accom_food_svc/total_pop,
#   occup_ind_other = occup_ind_other/total_pop,
```

```
#   occup_ind_pubadmin = occup_ind_pubadmin/total_pop,
#
#   work_loc_home = work_loc_home/total_pop,
#   work_loc_foreign = work_loc_foreign/total_pop,
#   work_loc_notfixed = work_loc_notfixed/total_pop,
#   work_loc_workplace = work_loc_workplace/total_pop,
#
#   commute_same_subdiv = commute_same_subdiv/total_pop,
#   commute_same_div = commute_same_div/total_pop,
#   commute_same_prov = commute_same_prov/total_pop,
#   commute_diff_prov = commute_diff_prov/total_pop,
#
#   commute_transp_cardriver = commute_transp_cardriver/total_pop,
#   commute_transp_carpass = commute_transp_carpass/total_pop,
#   commute_transp_pubtrans = commute_transp_pubtrans/total_pop,
#   commute_transp_walk = commute_transp_walk/total_pop,
#
#   commute_transp_bike = commute_transp_bike/total_pop,
#   commute_transp_other = commute_transp_other/total_pop,
#
#   commute_time_under15 = commute_time_under15/total_pop,
#   commute_time_15 = commute_time_15/total_pop,
#   commute_time_30 = commute_time_30/total_pop,
#   commute_time_45 = commute_time_45/total_pop,
#   commute_time_over60 = commute_time_over60/total_pop,
#
#   commute_start_5am = commute_start_5am/total_pop,
#   commute_start_6am = commute_start_6am/total_pop,
#   commute_start_7am = commute_start_7am/total_pop,
#   commute_start_8am = commute_start_8am/total_pop,
#   commute_start_9am = commute_start_9am/total_pop,
#   commute_start_noon = commute_start_noon/total_pop,
#   BEV = (BEV/total_pop)*10000,
#   PHEV = (PHEV/total_pop)*10000,
#   TotalEV = (TotalEV/total_pop)*10000
# )
#
# write.csv(new_data, "Data/fulldata_norm.csv")
```

EDA and Visualizations:

We will load the necessary libraries and the dataset for the initialization of the analysis.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(plotly)
```

```
## Warning: package 'plotly' was built under R version 4.3.3
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

```
## Loading required package: lattice
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.3.3
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.3
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

```
## corrplot 0.92 loaded
```

```
library(mgcv)
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
## This is mgcv 1.9-0. For overview type 'help("mgcv-package")'.
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3
```

```
## Warning: package 'readr' was built under R version 4.3.3
```

```
## Warning: package 'forcats' was built under R version 4.3.3
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## v readr     2.1.5
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x nlme::collapse()        masks dplyr::collapse()
## x randomForest::combine() masks dplyr::combine()
## x plotly::filter()        masks dplyr::filter(), stats::filter()
## x dplyr::lag()            masks stats::lag()
## x purrr::lift()           masks caret::lift()
## x randomForest::margin()  masks ggplot2::margin()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)

# Reading the data
data <- read.csv("data.csv")

# View the first 15 rows
head(data, 15)
```

```
##    FSA med_fulltime_income income_100k_up med_total_household_income income_40k
## 1  K0A               74500     0.12707613                      115000 0.07887947
## 2  K0B               56800     0.06351094                       79000 0.09800190
## 3  K0C               58800     0.06642946                       84000 0.09614293
## 4  K0E               58000     0.06355772                       83000 0.10252465
## 5  K0G               64500     0.09445086                       94000 0.09256936
## 6  K0H               64000     0.08289980                       92000 0.09030157
## 7  K0J               60000     0.07109143                       76000 0.08973836
## 8  K0K               58800     0.06905481                       83000 0.09676306
## 9  K0L               59200     0.06948888                       79000 0.10052213
## 10 K0M               58800     0.07442163                       80000 0.09895011
## 11 K1A                  NA     0.07575758                       73500 0.10942761
## 12 K1B               65500     0.08413529                       96000 0.08920029
## 13 K1C               78500     0.13441654                      117000 0.07385524
## 14 K1E               71000     0.10476978                      110000 0.08478081
## 15 K1G               68000     0.09066545                       83000 0.08325255
##    occup_cat_snr_mgmt occup_cat_busfin occup_ind_realestate work_loc_home
## 1         0.009092864       0.10423199          0.007256374    0.17065917
## 2         0.006422455       0.08396765          0.003805899    0.09990485
## 3         0.004542186       0.07579772          0.004826072    0.09131686
## 4         0.004161517       0.07011526          0.006557542    0.07805998
## 5         0.006647935       0.08090412          0.008905725    0.12493101
## 6         0.005392717       0.07021106          0.007507508    0.09685742
## 7         0.003933337       0.05011363          0.004224696    0.07196550
## 8         0.004488563       0.05774709          0.006387570    0.08446267
## 9         0.005054900       0.05982698          0.007102455    0.08362980
## 10        0.006132119       0.06317941          0.009198179    0.08287652
## 11        0.000000000       0.04208754          0.000000000    0.05892256
## 12        0.005346390       0.11143002          0.009285835    0.16292419
## 13        0.005505573       0.11937693          0.006848395    0.18812945
## 14        0.005858837       0.12234629          0.004824924    0.17266336
## 15        0.003706449       0.09280379          0.006985231    0.17334778
##    work_loc_workplace work_loc_notfixed school_uni_degree  school_hs
## 1          0.26763478        0.07534087        0.14777023 0.13186892
## 2          0.30566127        0.06874405        0.08087536 0.16841104
## 3          0.30669215        0.06557780        0.07948825 0.15074378
## 4          0.29962925        0.06973694        0.07490731 0.15473278
## 5          0.26428679        0.06898801        0.11150971 0.13897948
## 6          0.27153914        0.05910840        0.10066404 0.14782388
## 7          0.26659286        0.05841734        0.08012354 0.15121496
## 8          0.27449288        0.06283988        0.08165731 0.15248166
## 9          0.25255304        0.06136265        0.08593330 0.14895959
## 10         0.24036049        0.07209886        0.07795224 0.16092168
## 11         0.06734007        0.00000000        0.05892256 0.04208754
## 12         0.25099893        0.04530362        0.18768642 0.11649502
## 13         0.22008863        0.03880757        0.22814556 0.09748892
## 14         0.23021781        0.05204025        0.15991177 0.12475875
## 15         0.23265097        0.04461995        0.24020642 0.10477847
##    commute_start_noon edu_field_science edu_field_bus_admin edu_field_health
## 1          0.03847670       0.013840861          0.06915952       0.04779353
## 2          0.05042816       0.007136061          0.05233111       0.04567079
## 3          0.05743972       0.007948825          0.05138347       0.05214050
## 4          0.04527226       0.007061969          0.05536079       0.04880325
## 5          0.03838242       0.010034619          0.05945512       0.05017310
```

```
## 6          0.03373091       0.010891173          0.05170664       0.05530178
## 7          0.03714819       0.011800012          0.04661733       0.04894820
## 8          0.04760466       0.006862322          0.05058265       0.04777730
## 9          0.04331857       0.010173786          0.04549410       0.04926928
## 10         0.04394685       0.008176159          0.04561925       0.04171699
## 11         0.00000000       0.016835017          0.02525253       0.00000000
## 12         0.05599640       0.021948337          0.06894029       0.04136417
## 13         0.04095609       0.017859541          0.08392641       0.04726736
## 14         0.03791012       0.016542597          0.08064516       0.04824924
## 15         0.04490506       0.023379141          0.06999487       0.05046473
##    edu_field_pers_protect_transp commute_transp_carpass commute_start_6am
## 1                     0.02597961             0.02127641        0.09200366
## 2                     0.02664129             0.01760228        0.08610847
## 3                     0.02621220             0.02015595        0.09046520
## 4                     0.03430099             0.01841156        0.09823703
## 5                     0.02809693             0.02157443        0.08153128
## 6                     0.03235630             0.01628389        0.08194815
## 7                     0.02563953             0.02301731        0.08128897
## 8                     0.02507553             0.02063012        0.07699612
## 9                     0.02674618             0.01676435        0.06123468
## 10                    0.02304190             0.02071913        0.06039208
## 11                    0.00000000             0.00000000        0.01683502
## 12                    0.02166695             0.02729473        0.04952445
## 13                    0.01571102             0.02108232        0.04847590
## 14                    0.02412462             0.02171216        0.06203474
## 15                    0.01596624             0.02437703        0.04461995
##    commute_start_9am       condo worktype_selfemp indigenous_ppl married_ppl
## 1         0.02620357 0.026865126       0.08291079     0.04089549   0.5547543
## 2         0.02164605 0.037411740       0.09253092     0.03187441   0.5432921
## 3         0.02498202 0.013868099       0.07655475     0.03416102   0.5381544
## 4         0.02837398 0.008954738       0.07377235     0.03127443   0.5498247
## 5         0.02997843 0.015712740       0.08930811     0.03399227   0.5645728
## 6         0.02738654 0.003156317       0.08205388     0.04049825   0.5653893
## 7         0.02840744 0.005531808       0.06730377     0.09032108   0.5391586
## 8         0.03211049 0.011100032       0.08476478     0.06340095   0.5538196
## 9         0.03000947 0.008916492       0.08580533     0.06040286   0.5607740
## 10        0.03372666 0.012559457       0.08975193     0.02601505   0.5753972
## 11        0.00000000 0.108173077       0.03367003     0.01683502   0.4797980
## 12        0.04052001 0.308039615       0.05458945     0.03404806   0.4386854
## 13        0.02806499 0.215902739       0.05116154     0.02833356   0.5181952
## 14        0.02963882 0.159474672       0.04618142     0.02860491   0.5424593
## 15        0.04290928 0.122285267       0.05673718     0.02337914   0.4186862
##    work_loc_foreign can_citizen_ppl non_citizen_ppl    TotalEV
## 1       0.000806264       0.9712343      0.01567735   88.42026
## 2       0.000475737       0.9605138      0.01522360   56.61275
## 3       0.000851660       0.9672016      0.01343730   45.80037
## 4       0.001639386       0.9816137      0.01324119   43.63288
## 5       0.001630626       0.9700968      0.01455020   73.75445
## 6       0.000634437       0.9646618      0.01057395   61.96337
## 7       0.001602471       0.9681837      0.01150865   37.87658
## 8       0.000647389       0.9656021      0.01510574   48.51101
## 9       0.000959791       0.9673415      0.01414092   40.69514
## 10      0.001207842       0.9757503      0.01235715   34.00539
## 11      0.000000000       0.8080808      0.05050505 1414.14141
```

```
## 12      0.001406945       0.8686476      0.08582363   61.90557
## 13      0.001342823       0.9348731      0.04176178   82.98644
## 14      0.000689275       0.9642956      0.03239592   55.83127
## 15      0.001568113       0.8346639      0.13998974   67.00120
```

```r
# Getting information about the dataframe
str(data)
```

```
## 'data.frame':    518 obs. of  29 variables:
##  $ FSA                           : chr  "K0A" "K0B" "K0C" "K0E" ...
##  $ med_fulltime_income           : int  74500 56800 58800 58000 64500 64000 60000 58800 59200 58800 .
##  $ income_100k_up                : num  0.1271 0.0635 0.0664 0.0636 0.0945 ...
##  $ med_total_household_income    : int  115000 79000 84000 83000 94000 92000 76000 83000 79000 80000
##  $ income_40k                    : num  0.0789 0.098 0.0961 0.1025 0.0926 ...
##  $ occup_cat_snr_mgmt            : num  0.00909 0.00642 0.00454 0.00416 0.00665 ...
##  $ occup_cat_busfin              : num  0.1042 0.084 0.0758 0.0701 0.0809 ...
##  $ occup_ind_realestate          : num  0.00726 0.00381 0.00483 0.00656 0.00891 ...
##  $ work_loc_home                 : num  0.1707 0.0999 0.0913 0.0781 0.1249 ...
##  $ work_loc_workplace            : num  0.268 0.306 0.307 0.3 0.264 ...
##  $ work_loc_notfixed             : num  0.0753 0.0687 0.0656 0.0697 0.069 ...
##  $ school_uni_degree             : num  0.1478 0.0809 0.0795 0.0749 0.1115 ...
##  $ school_hs                     : num  0.132 0.168 0.151 0.155 0.139 ...
##  $ commute_start_noon            : num  0.0385 0.0504 0.0574 0.0453 0.0384 ...
##  $ edu_field_science             : num  0.01384 0.00714 0.00795 0.00706 0.01003 ...
##  $ edu_field_bus_admin           : num  0.0692 0.0523 0.0514 0.0554 0.0595 ...
##  $ edu_field_health              : num  0.0478 0.0457 0.0521 0.0488 0.0502 ...
##  $ edu_field_pers_protect_transp: num  0.026 0.0266 0.0262 0.0343 0.0281 ...
##  $ commute_transp_carpass        : num  0.0213 0.0176 0.0202 0.0184 0.0216 ...
##  $ commute_start_6am             : num  0.092 0.0861 0.0905 0.0982 0.0815 ...
##  $ commute_start_9am             : num  0.0262 0.0216 0.025 0.0284 0.03 ...
##  $ condo                         : num  0.02687 0.03741 0.01387 0.00895 0.01571 ...
##  $ worktype_selfemp              : num  0.0829 0.0925 0.0766 0.0738 0.0893 ...
##  $ indigenous_ppl                : num  0.0409 0.0319 0.0342 0.0313 0.034 ...
##  $ married_ppl                   : num  0.555 0.543 0.538 0.55 0.565 ...
##  $ work_loc_foreign              : num  0.000806 0.000476 0.000852 0.001639 0.001631 ...
##  $ can_citizen_ppl               : num  0.971 0.961 0.967 0.982 0.97 ...
##  $ non_citizen_ppl               : num  0.0157 0.0152 0.0134 0.0132 0.0146 ...
##  $ TotalEV                       : num  88.4 56.6 45.8 43.6 73.8 ...
```

We will now perform Missing Value analysis on the dataset.

```r
# Checking for rows with any missing values
#data[complete.cases(data), ]

# Subset the dataframe to include only rows with at least one NA value
rows_with_na <- data[apply(is.na(data), 1, any), ]

# View rows with NA values
print(rows_with_na)
```

```
##     FSA med_fulltime_income income_100k_up med_total_household_income
## 11  K1A                  NA     0.07575758                       73500
## 159 L4V                  NA             NA                          NA
```

```
## 176 L5S                  NA             NA                          NA
## 177 L5T                  NA             NA                          NA
##     income_40k occup_cat_snr_mgmt occup_cat_busfin occup_ind_realestate
## 11   0.1094276                  0       0.04208754                    0
## 159         NA                 NA               NA                   NA
## 176         NA                 NA               NA                   NA
## 177         NA                 NA               NA                   NA
##     work_loc_home work_loc_workplace work_loc_notfixed school_uni_degree
## 11     0.05892256         0.06734007                 0        0.05892256
## 159            NA                 NA                NA                NA
## 176            NA                 NA                NA                NA
## 177            NA                 NA                NA                NA
##      school_hs commute_start_noon edu_field_science edu_field_bus_admin
## 11  0.04208754                  0        0.01683502          0.02525253
## 159         NA                 NA                NA                  NA
## 176         NA                 NA                NA                  NA
## 177         NA                 NA                NA                  NA
##     edu_field_health edu_field_pers_protect_transp commute_transp_carpass
## 11                 0                             0                     0
## 159               NA                            NA                    NA
## 176               NA                            NA                    NA
## 177               NA                            NA                    NA
##     commute_start_6am commute_start_9am     condo worktype_selfemp
## 11         0.01683502                 0 0.1081731       0.03367003
## 159                NA                NA        NA               NA
## 176                NA                NA        NA               NA
## 177                NA                NA        NA               NA
##     indigenous_ppl married_ppl work_loc_foreign can_citizen_ppl non_citizen_ppl
## 11      0.01683502    0.479798                0       0.8080808      0.05050505
## 159             NA          NA               NA              NA              NA
## 176             NA          NA               NA              NA              NA
## 177             NA          NA               NA              NA              NA
##        TotalEV
## 11    1414.141
## 159 102000.000
## 176  22962.963
## 177  57368.421
```

```r
# Dropping rows by index
data <- data[-c(11, 159, 176, 177), ]
```

Upon analyzing the dataset rows, it becomes apparent that many values are missing and that the total number of electric vehicles is comparatively very high, classifying them as outliers. Consequently, we will proceed with their removal.

```r
# Subset the dataframe to include only rows with at least one NA value
rows_with_na <- data[apply(is.na(data), 1, any), ]

# View rows with NA values
print(rows_with_na)
```

```
##  [1] FSA                          med_fulltime_income
##  [3] income_100k_up               med_total_household_income
```

```
##  [5] income_40k                   occup_cat_snr_mgmt
##  [7] occup_cat_busfin             occup_ind_realestate
##  [9] work_loc_home                work_loc_workplace
## [11] work_loc_notfixed            school_uni_degree
## [13] school_hs                    commute_start_noon
## [15] edu_field_science            edu_field_bus_admin
## [17] edu_field_health             edu_field_pers_protect_transp
## [19] commute_transp_carpass       commute_start_6am
## [21] commute_start_9am            condo
## [23] worktype_selfemp             indigenous_ppl
## [25] married_ppl                  work_loc_foreign
## [27] can_citizen_ppl              non_citizen_ppl
## [29] TotalEV
## <0 rows> (or 0-length row.names)
```

```r
# Checking for NA values
colSums(is.na(data))
```

```
##                          FSA           med_fulltime_income
##                            0                             0
##               income_100k_up    med_total_household_income
##                            0                             0
##                   income_40k           occup_cat_snr_mgmt
##                            0                             0
##             occup_cat_busfin         occup_ind_realestate
##                            0                             0
##                work_loc_home           work_loc_workplace
##                            0                             0
##            work_loc_notfixed            school_uni_degree
##                            0                             0
##                    school_hs           commute_start_noon
##                            0                             0
##            edu_field_science          edu_field_bus_admin
##                            0                             0
##             edu_field_health edu_field_pers_protect_transp
##                            0                             0
##       commute_transp_carpass            commute_start_6am
##                            0                             0
##            commute_start_9am                         condo
##                            0                             0
##             worktype_selfemp               indigenous_ppl
##                            0                             0
##                 married_ppl             work_loc_foreign
##                            0                             0
##              can_citizen_ppl              non_citizen_ppl
##                            0                             0
##                      TotalEV
##                            0
```

We hereby confirm that there are no missing values in all the represented features.

Descriptive Statistics

```r
build_continuous_features_report <- function(data_df) {
  # Selecting continuous (numeric) features
  continuous_data_df <- data_df[sapply(data_df, is.numeric)]

  # Function to calculate statistics for a single column
  stats <- function(df) {
    data.frame(
      "Count" = length(df),
      "Miss %" = sum(is.na(df)) / length(df) * 100,
      "Card." = length(unique(na.omit(df))),
      "Min" = min(df, na.rm = TRUE),
      "1st Qrt." = quantile(df, 0.25, na.rm = TRUE),
      "Mean" = mean(df, na.rm = TRUE),
      "Median" = median(df, na.rm = TRUE),
      "3rd Qrt" = quantile(df, 0.75, na.rm = TRUE),
      "Max" = max(df, na.rm = TRUE),
      "Std. Dev." = sd(df, na.rm = TRUE)
    )
  }

  # Initializing an empty list to store the statistics of each feature
  report_list <- list()

  # Looping through each continuous feature to calculate the statistics
  for(feature in names(continuous_data_df)) {
    feature_data <- continuous_data_df[[feature]]
    report_list[[feature]] <- t(stats(feature_data)) # Transpose for correct orientation
  }

  # Combining the individual feature reports into a single dataframe
  report_df <- do.call(cbind, report_list)
  colnames(report_df) <- names(continuous_data_df)

  return(t(report_df)) # Transpose to match desired output format
}

# Example of how to use the function with a dataframe `data`
result <- build_continuous_features_report(data)
print(result)
```

```
##                               Count Miss.. Card.          Min    X1st.Qrt.
## med_fulltime_income             514      0    97 4.440000e+04 5.880000e+04
## income_100k_up                  514      0   514 1.273074e-02 6.141190e-02
## med_total_household_income      514      0   121 4.520000e+04 7.700000e+04
## income_40k                      514      0   514 4.717379e-02 7.340043e-02
## occup_cat_snr_mgmt              514      0   507 0.000000e+00 3.388454e-03
## occup_cat_busfin                514      0   514 3.297573e-02 6.900668e-02
## occup_ind_realestate            514      0   513 0.000000e+00 6.928390e-03
## work_loc_home                   514      0   514 2.331764e-02 7.955200e-02
## work_loc_workplace              514      0   513 1.674410e-01 2.403227e-01
## work_loc_notfixed               514      0   514 1.712542e-02 4.356903e-02
## school_uni_degree               514      0   514 4.198352e-02 1.089500e-01
## school_hs                       514      0   513 1.550388e-02 1.033002e-01
```

```
## commute_start_noon              514      0   514 1.137656e-02 3.982945e-02
## edu_field_science               514      0   514 2.772497e-03 9.195521e-03
## edu_field_bus_admin             514      0   514 2.725994e-02 5.677883e-02
## edu_field_health                514      0   514 2.275313e-02 4.423606e-02
## edu_field_pers_protect_transp   514      0   511 0.000000e+00 1.490709e-02
## commute_transp_carpass          514      0   513 5.306373e-03 1.951939e-02
## commute_start_6am               514      0   514 0.000000e+00 4.430966e-02
## commute_start_9am               514      0   514 2.050441e-02 3.522463e-02
## condo                           514      0   498 0.000000e+00 3.200050e-02
## worktype_selfemp                514      0   514 2.418933e-02 5.555032e-02
## indigenous_ppl                  514      0   510 0.000000e+00 9.150866e-03
## married_ppl                     514      0   514 3.193406e-01 4.460952e-01
## work_loc_foreign                514      0   440 0.000000e+00 7.800400e-04
## can_citizen_ppl                 514      0   514 6.266024e-01 8.544114e-01
## non_citizen_ppl                 514      0   513 0.000000e+00 2.514110e-02
## TotalEV                         514      0   514 4.320276e+00 4.072137e+01
##                                       Mean       Median     X3rd.Qrt
## med_fulltime_income           6.813911e+04 6.550000e+04 7.550000e+04
## income_100k_up                1.001158e-01 8.636316e-02 1.289127e-01
## med_total_household_income    9.425914e+04 9.000000e+04 1.100000e+05
## income_40k                    8.438633e-02 8.491067e-02 9.658390e-02
## occup_cat_snr_mgmt            7.241126e-03 5.386459e-03 8.918425e-03
## occup_cat_busfin              9.135088e-02 8.680438e-02 1.093488e-01
## occup_ind_realestate          1.047029e-02 9.235419e-03 1.273603e-02
## work_loc_home                 1.326125e-01 1.126607e-01 1.703045e-01
## work_loc_workplace            2.681378e-01 2.687989e-01 2.980605e-01
## work_loc_notfixed             5.282995e-02 5.147544e-02 6.135622e-02
## school_uni_degree             1.897277e-01 1.702661e-01 2.458678e-01
## school_hs                     1.252665e-01 1.273340e-01 1.497220e-01
## commute_start_noon            5.125881e-02 5.143437e-02 6.210672e-02
## edu_field_science             1.506847e-02 1.449348e-02 1.972872e-02
## edu_field_bus_admin           7.863199e-02 7.194441e-02 9.478226e-02
## edu_field_health              5.089910e-02 4.895080e-02 5.609437e-02
## edu_field_pers_protect_transp 1.960686e-02 1.969967e-02 2.487012e-02
## commute_transp_carpass        2.355430e-02 2.297626e-02 2.829750e-02
## commute_start_6am             5.670137e-02 5.719879e-02 6.984864e-02
## commute_start_9am             4.305803e-02 4.172278e-02 4.901688e-02
## condo                         1.240767e-01 8.277550e-02 1.654754e-01
## worktype_selfemp              7.521060e-02 6.945875e-02 8.967989e-02
## indigenous_ppl                3.652714e-02 1.952549e-02 3.564316e-02
## married_ppl                   4.781837e-01 4.817845e-01 5.142772e-01
## work_loc_foreign              2.146508e-03 1.373440e-03 2.278080e-03
## can_citizen_ppl               8.981905e-01 9.166494e-01 9.549297e-01
## non_citizen_ppl               8.637394e-02 6.765510e-02 1.288833e-01
## TotalEV                       7.856578e+01 6.500723e+01 9.992947e+01
##                                        Max    Std..Dev.
## med_fulltime_income           1.180000e+05 1.245777e+04
## income_100k_up                2.823795e-01 5.334634e-02
## med_total_household_income    1.980000e+05 2.348615e+04
## income_40k                    1.176626e-01 1.476160e-02
## occup_cat_snr_mgmt            4.550626e-02 6.259000e-03
## occup_cat_busfin              2.759148e-01 3.145830e-02
## occup_ind_realestate          3.379274e-02 5.524704e-03
## work_loc_home                 4.806202e-01 7.300509e-02
```

```
## work_loc_workplace            4.275362e-01 4.046317e-02
## work_loc_notfixed             1.304945e-01 1.485901e-02
## school_uni_degree             5.988300e-01 1.036152e-01
## school_hs                     1.989197e-01 3.329437e-02
## commute_start_noon            1.035937e-01 1.603235e-02
## edu_field_science             6.101938e-02 7.598107e-03
## edu_field_bus_admin           2.776127e-01 3.018230e-02
## edu_field_health              1.199877e-01 1.101826e-02
## edu_field_pers_protect_transp 3.552014e-02 6.665267e-03
## commute_transp_carpass        4.357299e-02 7.071020e-03
## commute_start_6am             1.009389e-01 1.876674e-02
## commute_start_9am             9.684361e-02 1.153010e-02
## condo                         8.502848e-01 1.385631e-01
## worktype_selfemp              2.218430e-01 2.814755e-02
## indigenous_ppl                7.006189e-01 6.126543e-02
## married_ppl                   6.555469e-01 5.441609e-02
## work_loc_foreign              3.529948e-02 3.361950e-03
## can_citizen_ppl               1.075405e+00 7.131559e-02
## non_citizen_ppl               3.532971e-01 7.092823e-02
## TotalEV                       5.426357e+02 5.642360e+01
```

1. **Data Completeness**: For most of the variables, the count is 514, indicating that the dataset is relatively complete with few missing values (denoted by "Miss" column). However, some variables do have missing data (e.g., 'occup_ind_realestate', 'school_hs', 'commute_transp_carpass', and a few others).

2. **Income Distribution**: The `med_fulltime_income` has a mean of around $63,819 with a median of $65,500, which suggests a somewhat symmetrical distribution around the median income but there are also high-income earners as evidenced by a maximum value of around $118,000. The standard deviation is quite large, indicating a wide spread in full-time income.

3. **High Income Earners**: The `income_100k_up` variable suggests that about 1.27% of the observations represent people earning above $100,000, as its mean is close to 0.012731, which matches the proportion since the variable likely indicates a binary (0 or 1) for individuals in this income bracket.

4. **Housing**: The `condo` variable suggests that approximately 2% of the observations are associated with condominiums. The nature of the variable is not clear, but if it's binary, it may indicate whether the person lives in a condo or not.

5. **Education**: A small percentage (around 1.98%) have a university degree (`school_uni_degree`), while a slightly smaller percentage have completed high school (`school_hs`).

6. **Occupation**: There's a very small percentage of people in senior management (`occup_cat_snr_mgmt`) and business finance (`occup_cat_busfin`). The `occup_ind_realestate` might indicate those involved in real estate, which is also a small portion of the population.

7. **Work Location**: A fair number of people work from home (`work_loc_home`), as indicated by a mean of around 0.06.

8. **Commute Times**: Few people start their commute at noon or at 6 am (`commute_start_noon`, `commute_start_6am`), as these values are close to zero.

9. **Diversity**: The variables `indigenous_ppl`, `married_ppl`, `worktype_selfemp`, and `non_citizen_ppl` likely indicate proportions or counts of these demographics within the dataset.

10. **TotalEV**: It's unclear what this variable represents as it's not labeled like the others, but it has a mean of around 4.32 and a very high standard deviation of 56.42, indicating a very wide distribution or range of values.

Categorical Feature Report:

```
describe_categorical <- function(df) {
  # Identify non-numeric (categorical and character) columns
  cat_cols <- df[sapply(df, function(column) !is.numeric(column))]

  # Initialize a list to store summaries
  summaries <- list()

  # Loop through each categorical column to calculate summary statistics
  for (col_name in names(cat_cols)) {
    col_data <- na.omit(cat_cols[[col_name]])  # Remove NAs for accurate calculations

    # Calculate summary statistics
    summary_stats <- list(
      "Count" = length(col_data),
      "Unique" = length(unique(col_data)),
      "Top" = names(sort(table(col_data), decreasing = TRUE)[1]),
      "Freq" = sort(table(col_data), decreasing = TRUE)[[1]]
    )

    # Add summary statistics to the list
    summaries[[col_name]] <- summary_stats
  }

  # Convert the list of summaries into a readable format, e.g., a data frame
  summary_df <- do.call(rbind, lapply(summaries, function(x) do.call(cbind, x)))
  rownames(summary_df) <- names(summaries)

  return(summary_df)
}

# Example usage with a dataframe 'df'
summary_df <- describe_categorical(data)
print(summary_df)
```

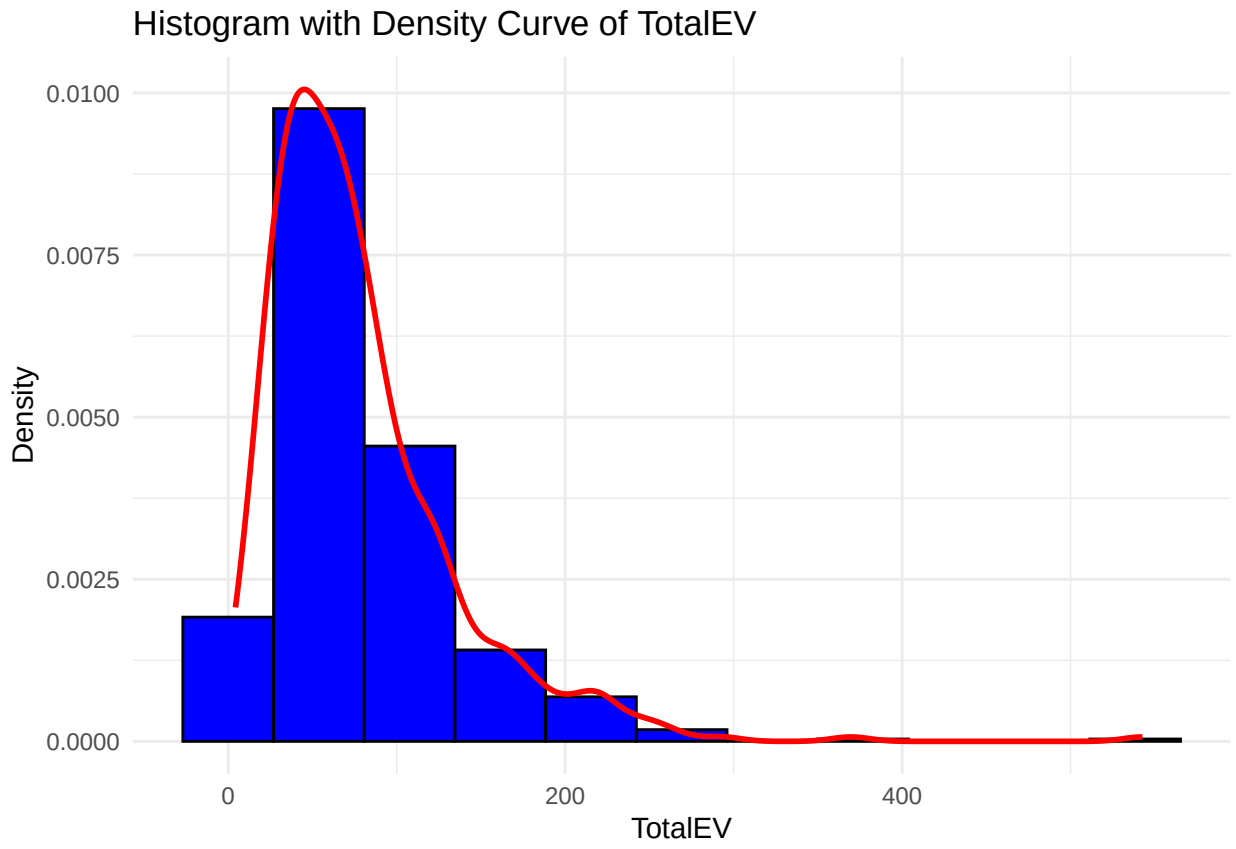
```
##     Count Unique Top   Freq
## FSA "514" "514"  "KOA" "1"
```

This is a summary report of categorical features from a dataframe. It shows two categorical columns: 'FSA' and 'TotalEV_Category'. For 'FSA', there are 514 unique categories (Cardinality), with the most frequent category (Mode) occurring 514 times, indicating all entries might be unique. For 'TotalEV_Category', there are 514 counts with 3 unique categories, the most common being 'Medium' with a frequency of 202, and the second most common 'Low' with a frequency of 183.

Before proceeding with further modelling and analysis, it is important to note the trends followed by the response variable.

```
ggplot(data, aes(x = TotalEV)) +
  geom_histogram(aes(y = ..density..), binwidth = diff(range(data$TotalEV)) / 10, color = "black", fill
  geom_density(adjust=1, color="red", size=1) +  # Adding the density curve
  labs(x = "TotalEV", y = "Density", title = "Histogram with Density Curve of TotalEV") +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.

## Warning: The dot-dot notation (‘..density..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(density)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



1. **Right-Skewed Distribution**: The histogram shows that most of the data points for the variable `TotalEV` are clustered towards the left side of the graph, with a long tail extending to the right. This indicates a right-skewed (or positively skewed) distribution.

2. **Majority of Low Values**: The height of the bars in the histogram indicates that the majority of the values for `TotalEV` are low, with a high frequency of values close to zero.

3. **Few High Values**: There are very few high values, as seen by the presence of bars to the right of the graph that are much shorter in height. This is typical of a skewed distribution where a small number of observations are much larger than the rest.

4. **Possible Outliers**: The long tail to the right suggests there could be outliers or extreme values in the `TotalEV` data. These could be unusually high values relative to the rest of the data.

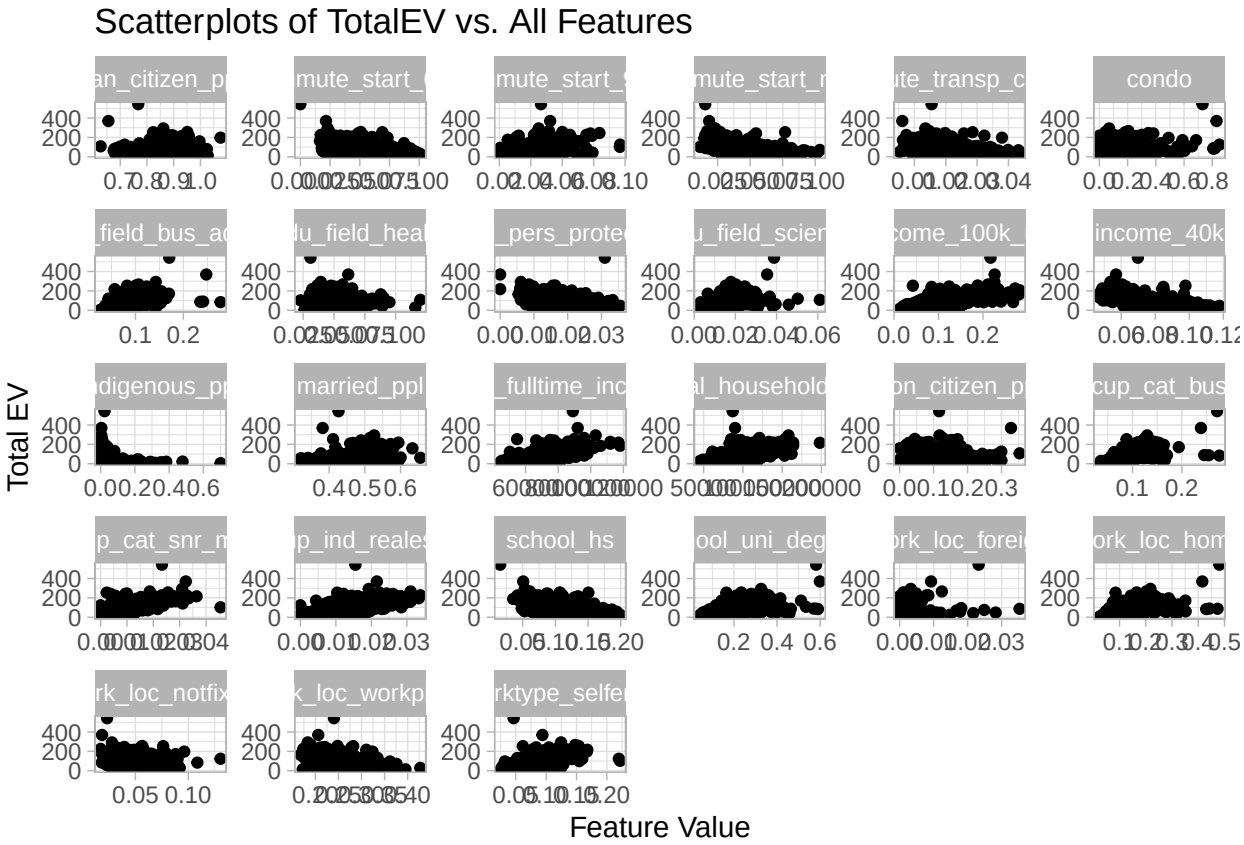
5. **Density Curve**: The KDE line tries to estimate a smooth distribution of the data. It peaks where the histogram bars are the tallest, reflecting the concentration of data points. The long right tail of the KDE suggests that while there are few large values, they extend quite far from the most frequent values.

6. **Central Tendency**: Given the skewness of the data, the mean of `TotalEV` will be higher than the median, which is also suggested by the previous statistical summary provided, where the mean is greater than the median.

Essentially, the behavior of the response variable has prompted us to adopt a different perspective, as it is clear from the plot above that the distribution might not be normal. Therefore, it is necessary to further investigate the explanatory variables, fit a model, and then evaluate its statistical significance.

We will check how different explanatory variables have an effect with our response variable using scatter plots.

```r
# Load necessary libraries
library(ggplot2)
library(tidyr)


# Transform data from wide to long format
data_long <- pivot_longer(data, cols = -c(TotalEV, FSA), names_to = "Feature", values_to = "Value")

# Plot
ggplot(data_long, aes(x = Value, y = TotalEV)) +
  geom_point() +
  facet_wrap(~Feature, scales = "free") +
  theme_light() +
  labs(x = "Feature Value", y = "Total EV", title = "Scatterplots of TotalEV vs. All Features")
```



From the image of the scatter plots provided, here are some insights we could infer:

1. **Income and TotalEV Relationship**: The scatter plots generally indicate that there is a positive relationship between median full-time income, median total household income, and the proportion of income above $100k and $40k, with the 'TotalEV' metric. This suggests that as income increases, the 'TotalEV' also tends to be higher.

2. **Concentration of Categories**: For all the plots, the 'Low' and 'Medium' TotalEV categories seem to be more densely packed and primarily found in the lower range of the respective income metrics. The 'High' TotalEV category points, while less frequent, appear across a wider range of income values but tend to cluster more towards the higher end of the income metrics.

3. **Senior Management Occupation**: For the scatter plot with 'occup_cat_snr_mgmt' (senior management occupation), it looks like the higher the percentage of individuals in senior management, the higher the 'TotalEV'. However, there's a lot of overlap between the 'Low' and 'Medium' categories.

4. **Business and Finance Occupation**: Similar to senior management, the 'occup_cat_busfin' (business and finance occupation) scatter plot shows that areas with a higher percentage in these occupations might have higher 'TotalEV', but again with significant overlap in 'Low' and 'Medium' categories.

5. **Real Estate Occupation**: The 'occup_ind_realestate' scatter plot suggests a clear correlation with 'TotalEV' compared to full-time and household income, with a relatively even spread across the 'TotalEV' categories, although still with a tendency for higher 'TotalEV' at higher percentages in real estate occupations.

6. **Work from Home**: The 'work_loc_home' plot is interesting; it shows a concentration of 'High' TotalEV values at the lower range of the 'work_loc_home' metric. This could indicate that the ability to work from home is not as strongly associated with higher 'TotalEV' as the other financial metrics.

7. **Outliers**: There are a few outlier points with very high 'TotalEV' values across the plots. These outliers could represent specific areas or circumstances where 'TotalEV' is particularly high and are worth investigating further.

8. **Data Spread**: There's a notable spread of the 'TotalEV' values at higher income levels in several plots, which suggests that while income may be a good indicator of 'TotalEV', there are likely other factors at play influencing the 'TotalEV'.

9. **Real Estate Variables**: Variables such as 'work_loc_home' and 'condo' indicate a relationship with 'TotalEV'. It's possible that regions with higher work-from-home rates and more condos have higher 'TotalEV', but this might reflect a broader socioeconomic status.

10. **Education Variables**: There are plots for 'school_uni_degree', 'school_hs', 'edu_field_science', 'edu_field_bus_admin', and 'edu_field_health'. These suggest varying degrees of correlation with 'TotalEV'. For instance, areas with higher proportions of university degrees or certain fields of study might have higher 'TotalEV'.

11. **Commuting Variables**: The variables related to commuting, such as 'commute_start_6am', 'commute_start_9am', and 'commute_start_noon', might reflect lifestyle or regional differences that correlate with 'TotalEV'.

12. **Demographic Variables**: Scatter plots for 'indigenous_ppl', 'married_ppl', 'work_loc_foreign', 'can_citizen_ppl', and 'non_citizen_ppl' indicate how different demographic factors might be associated with 'TotalEV'. For example, regions with a higher proportion of married people or Canadian citizens may have different 'TotalEV' profiles.

13. **Distribution of TotalEV Categories**: The distribution of 'Low', 'Medium', and 'High' TotalEV categories varies across different variables, indicating complex socioeconomic factors at play.

14. **Density and Clustering**: There is a noticeable density of data points in certain ranges for each variable, with clustering around certain trends. This could inform targeted analyses or interventions.

These plots could serve as a basis for further analysis, such as looking into what makes the 'TotalEV' areas unique or understanding the role of occupation type in 'TotalEV'. It's also evident that income is an important but not sole factor in determining 'TotalEV', pointing to the multifaceted nature of this metric.

Research Question 1

We will begin by fitting a GLM to our data firstly because the Generalized Linear Model (GLM) is a way to make sense of data and relationships between variables in a more flexible way compared to traditional linear models.

```
data$TotalEV <- ceiling(data$TotalEV)
data$TotalEV
```

```
##   [1]  89  57  46  44  74  62  38  49  41  35  62  83  56  68 118  86  70  48
##  [19] 190  57 543  53 151  71  73 112 112 132  80 153  54  67  81  76  87  87
##  [37] 126  66 107  61 219 115 131 142 158  82 141 177 106 204 158  93  52  37
##  [55]  26  27  29  39  30  67  45  77  52 121  59  41  71  39  45  31  27  47
##  [73]  45  28  80  42  70  41  48  66  61  27  85  77  76  52  98 123 217  63
##  [91]  97  60  84 205  92  98  65  71  67  61  38  58  41  70 105 102  77 124
## [109]  76  84  83  89  89 102 198 116  46  38  53  64  53  31  62  46  96  57
## [127]  51  54  72  39  44  40 103 132 127 166  75 124  50 134  98 105 162 246
## [145] 125 173 170 138 120 242  98  61  67  76  42 174  41 223  44  75  82  34
## [163]  63  87  88 155 219 102 101  99 107  85 104  88 122 123 162 241 135 126
```

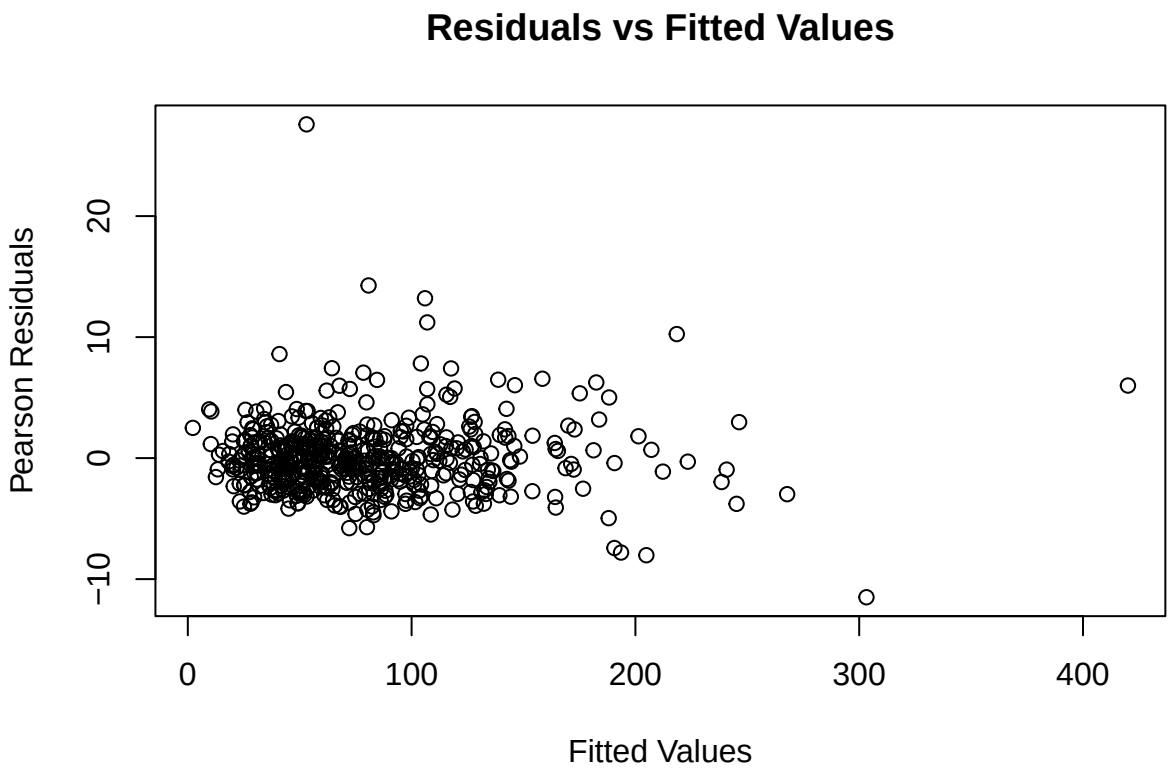
```
## [181] 191 293 209 219 180 113  68  58  64  45  65  84  90  64  77 257 184 112
## [199] 115 102 153 138 142 148 126 144  96 120 125  86  62  33  86  41  30  51
## [217]  45  68  34  58  40  29  50  48  75  64 182 150 141 172 165  75  46 166
## [235] 108  69  92 110  70  85 129  94  55  58  99  46  30  54  22  23  37  69
## [253]  76  37  44  58  42  44  41  79  79  73 103 227  88 106 227  47 117 267
## [271]  51  78  56  77  24  19  14  36  61  47 106 185  17  63  77  92 102 226
## [289]  74 165  89 165 143 196  23  40  65  35 172  85 108 370  90 186 130 162
## [307] 121  63  51  88  43  63 102  35  70  42  72  54  29  23  30  60  99 124
## [325]  82  87 208 115 215 113  99  60  63  25  38  46  30  33 254  52 117  58
## [343]  44  37  77  48  30  66  48  36  32  72  30 120  72  98  73  71 141  64
## [361]  68  48  81  83  79  73  50  45  81  85  90 129  82  54  60 121 113 121
## [379] 127  84 121  66  98 140  56  90  41  44  38  77  97  69  38  22  50  60
## [397]  52  44  48  67  64  50  64  81  59  46  29  65  41  41  28  29  68  20
## [415]  28  71  42  54  28  72  50  32  85  91  56 124 112  63  41  28  37  31
## [433]  21  32  63 160  22  31  43  70  58  38  32  19  35  33  28  29  25  18
## [451]  42  49  87  76  95  49  62  38  69  83  59  31  33  22   9  14  29   8
## [469]  22  13  17  18   6   7  23  19  30  84  55  54  43  41  26   5  32  32
## [487]  17  64 132  22  33  56  35  14  13  34   6  12  25  21  20  10  23  31
## [505]  18  16  43  42  23  23  17  15  10  20
```

```r
# Fit a GLM with a Poisson distribution
glm_poisson_model <- glm(TotalEV ~ med_fulltime_income + income_100k_up +
                           med_total_household_income + income_40k +
                           occup_cat_snr_mgmt + occup_cat_busfin +
                           occup_ind_realestate + work_loc_home +
                           work_loc_workplace + work_loc_notfixed +
                           school_uni_degree + school_hs +
                           commute_start_noon + edu_field_science +
                           edu_field_bus_admin + edu_field_health +
                           edu_field_pers_protect_transp + commute_transp_carpass +
                           commute_start_6am + commute_start_9am +
                           condo + worktype_selfemp + indigenous_ppl +
                           married_ppl + work_loc_foreign +
                           can_citizen_ppl + non_citizen_ppl,
                         data = data, family = poisson())

# Summary of the model
summary(glm_poisson_model)
```

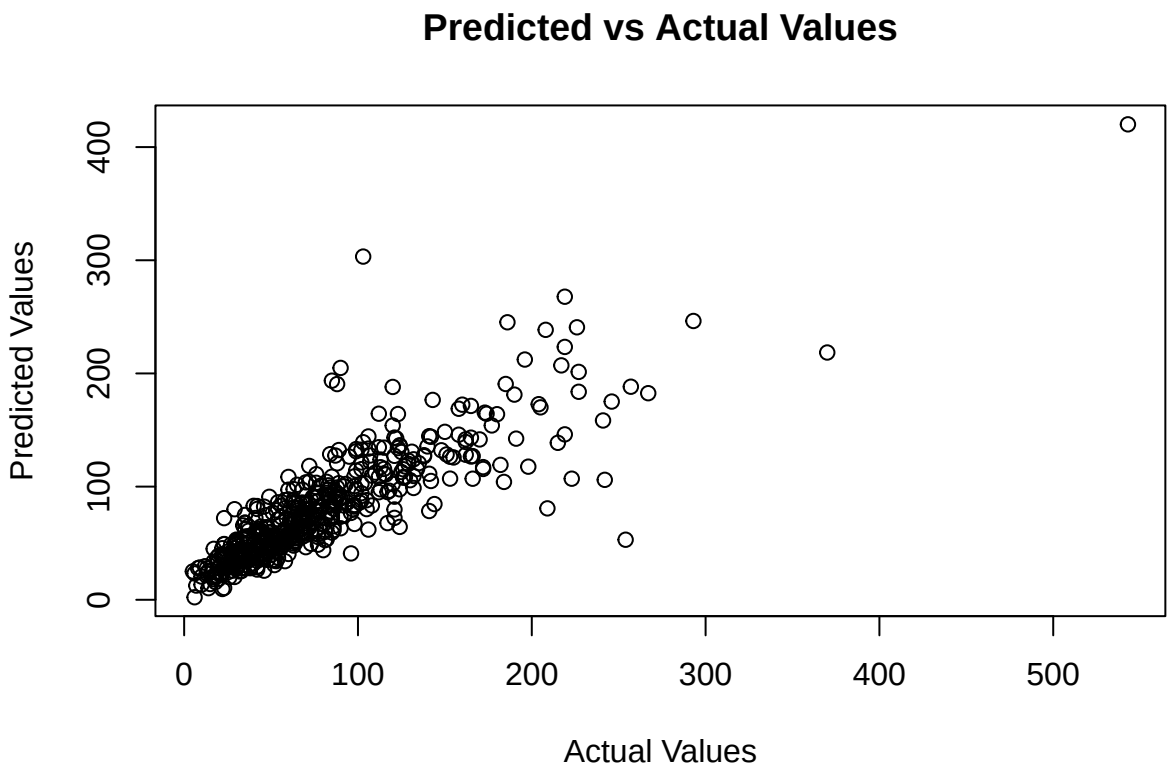
```
##
## Call:
## glm(formula = TotalEV ~ med_fulltime_income + income_100k_up +
##     med_total_household_income + income_40k + occup_cat_snr_mgmt +
##     occup_cat_busfin + occup_ind_realestate + work_loc_home +
##     work_loc_workplace + work_loc_notfixed + school_uni_degree +
##     school_hs + commute_start_noon + edu_field_science + edu_field_bus_admin +
##     edu_field_health + edu_field_pers_protect_transp + commute_transp_carpass +
##     commute_start_6am + commute_start_9am + condo + worktype_selfemp +
##     indigenous_ppl + married_ppl + work_loc_foreign + can_citizen_ppl +
##     non_citizen_ppl, family = poisson(), data = data)
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   4.888e+00  3.165e-01  15.445  < 2e-16 ***
```

```
## med_fulltime_income            7.213e-06  1.543e-06   4.674 2.95e-06 ***
## income_100k_up                -3.187e+00  4.484e-01  -7.107 1.19e-12 ***
## med_total_household_income     4.399e-06  6.289e-07   6.995 2.64e-12 ***
## income_40k                    -2.333e+00  1.065e+00  -2.191 0.028468 *
## occup_cat_snr_mgmt             2.099e+01  1.835e+00  11.441  < 2e-16 ***
## occup_cat_busfin              -2.078e+00  6.570e-01  -3.163 0.001560 **
## occup_ind_realestate           1.849e+01  1.869e+00   9.894  < 2e-16 ***
## work_loc_home                  1.873e+00  3.763e-01   4.977 6.47e-07 ***
## work_loc_workplace            -4.301e-01  3.943e-01  -1.091 0.275394
## work_loc_notfixed              7.035e+00  6.274e-01  11.213  < 2e-16 ***
## school_uni_degree              7.769e-01  3.571e-01   2.175 0.029599 *
## school_hs                     -2.577e+00  4.919e-01  -5.240 1.61e-07 ***
## commute_start_noon             3.307e+00  8.325e-01   3.972 7.12e-05 ***
## edu_field_science              8.140e+00  1.373e+00   5.928 3.07e-09 ***
## edu_field_bus_admin            1.996e+00  5.460e-01   3.656 0.000256 ***
## edu_field_health              -1.032e+00  6.352e-01  -1.625 0.104240
## edu_field_pers_protect_transp -3.207e+00  1.484e+00  -2.161 0.030678 *
## commute_transp_carpass         4.461e+00  1.327e+00   3.363 0.000771 ***
## commute_start_6am             -3.451e-03  8.715e-01  -0.004 0.996840
## commute_start_9am              3.608e+00  9.269e-01   3.892 9.93e-05 ***
## condo                          5.016e-01  6.448e-02   7.779 7.30e-15 ***
## worktype_selfemp              -2.006e+00  4.617e-01  -4.345 1.40e-05 ***
## indigenous_ppl                -3.972e+00  2.289e-01 -17.357  < 2e-16 ***
## married_ppl                    3.428e+00  2.195e-01  15.620  < 2e-16 ***
## work_loc_foreign               1.019e+01  1.543e+00   6.603 4.02e-11 ***
## can_citizen_ppl               -3.470e+00  3.369e-01 -10.302  < 2e-16 ***
## non_citizen_ppl               -5.414e+00  3.810e-01 -14.212  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 17307.1  on 513  degrees of freedom
## Residual deviance:  4127.4  on 486  degrees of freedom
## AIC: 7260.3
##
## Number of Fisher Scoring iterations: 4
```

```r
#Residual plots
# Residuals vs Fitted Values
plot(glm_poisson_model$fitted.values, residuals(glm_poisson_model, type = "pearson"),
     xlab = "Fitted Values", ylab = "Pearson Residuals",
     main = "Residuals vs Fitted Values")
```




```r
# Predicted vs Actual Values
predicted_values <- predict(glm_poisson_model, type = "response")
actual_values <- data$TotalEV

plot(actual_values, predicted_values,
     xlab = "Actual Values", ylab = "Predicted Values",
     main = "Predicted vs Actual Values")
```




```
# If necessary, install the coefplot package
if (!require("coefplot")) install.packages("coefplot")
```

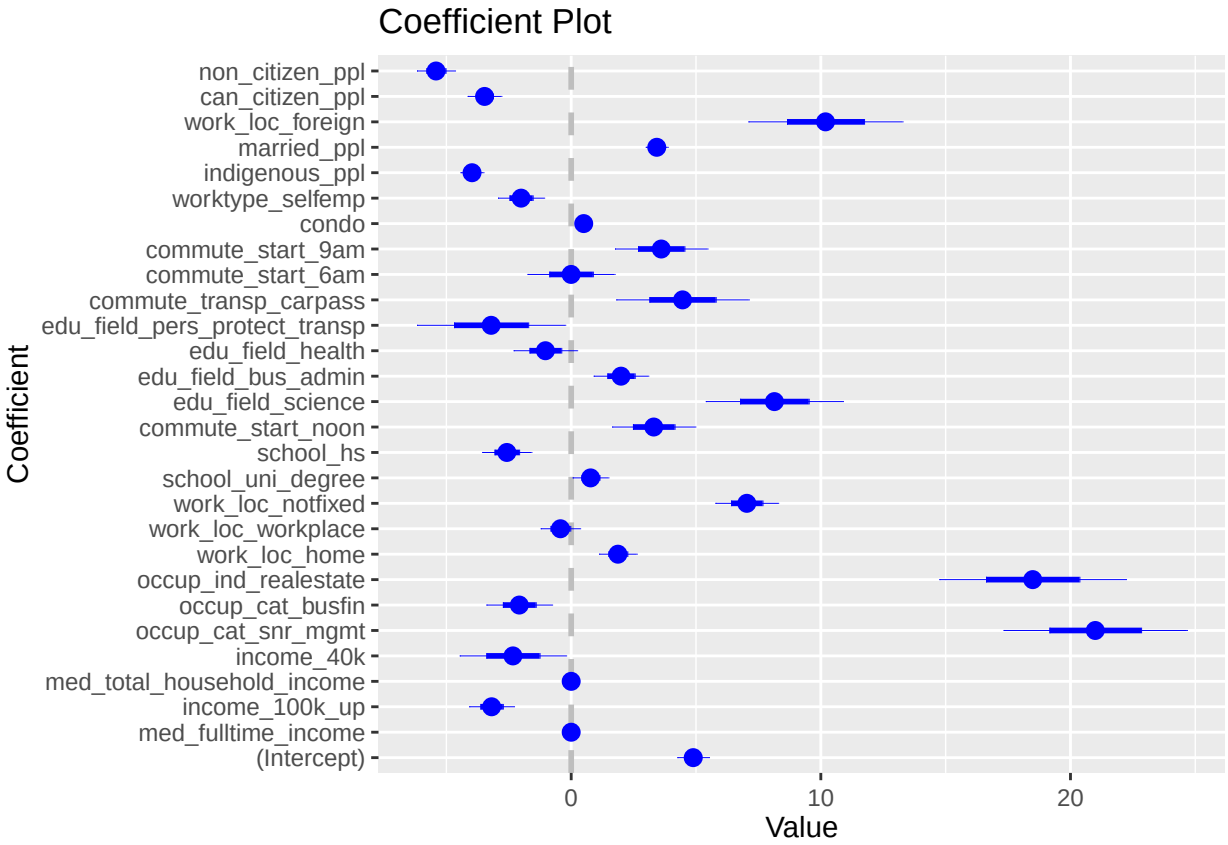
```
## Loading required package: coefplot
```

```
## Warning: package 'coefplot' was built under R version 4.3.3
```

```
##
## Attaching package: 'coefplot'
```

```
## The following object is masked from 'package:e1071':
##
##     extractPath
```

```
library(coefplot)

# Plotting model coefficients
coefplot(glm_poisson_model)
```



The residual plots we've provided offer valuable insights:

1. **Residuals vs Fitted Values Plot**: Ideally, in this plot, we'd want to see a random scatter of points with no discernible pattern, and the residuals roughly forming a horizontal band around the zero line. The clear pattern in our plot, with residuals increasing as the fitted values increase, might indicate a systematic variation that's not captured by the model. This is a typical sign of overdispersion or perhaps that the mean-variance relationship assumed by the Poisson model does not hold.
2. **Predicted vs Actual Values Plot**: The predicted values seem to be underestimating the actual values, especially as the actual values increase. This further indicates that the Poisson model might not be the correct model for our data, as it appears unable to capture the variance adequately, which seems to be increasing with the mean.
3. **Coefficient Plot**: This plot indicates the estimated effect sizes and their confidence intervals. The width of the confidence intervals indicates the precision of the estimates. A narrow interval means more precision, whereas a wide interval indicates less precision. Some variables have wide confidence intervals, which may be a result of overdispersion affecting the standard errors of the estimates.

Considering the evidence from the model summary and the diagnostic plots, it seems likely that the Poisson model is not adequately fitting our data due to overdispersion. The model is likely underestimating the variance, leading to too narrow confidence intervals and p-values that might be too optimistic.

Given these findings, it would be reasonable to consider alternative models that can account for overdispersion, such as:

- **Quasi-Poisson GLM**: This model adjusts for overdispersion by estimating a dispersion parameter from the data, which is used to correct standard errors. It is a quick fix that can be appropriate if the overdispersion is not severe.

- **Negative Binomial GLM**: This model has an additional parameter to explicitly model the overdispersion. It's generally more flexible and is recommended when there's strong evidence of overdispersion.

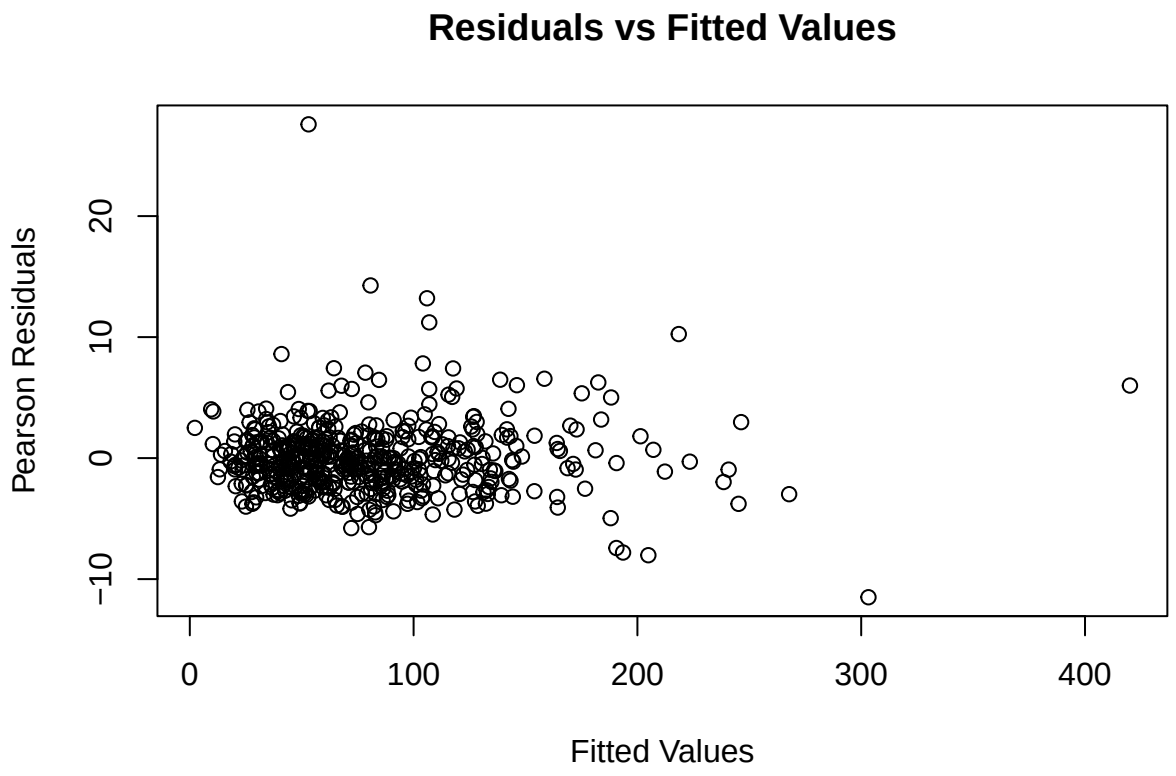
Trying both models and comparing their AIC values and diagnostic plots would be a good approach. Lower AIC values indicate better model fit when comparing models that use the same dataset. The choice between these models could also depend on the nature of the data and the underlying processes we're trying to model.

```
# Fit a GLM with a Quasi Poisson distribution
glm_quasi_poisson_model <- glm(TotalEV ~ med_fulltime_income + income_100k_up +
                           med_total_household_income + income_40k +
                           occup_cat_snr_mgmt + occup_cat_busfin +
                           occup_ind_realestate + work_loc_home +
                           work_loc_workplace + work_loc_notfixed +
                           school_uni_degree + school_hs +
                           commute_start_noon + edu_field_science +
                           edu_field_bus_admin + edu_field_health +
                           edu_field_pers_protect_transp + commute_transp_carpass +
                           commute_start_6am + commute_start_9am +
                           condo + worktype_selfemp + indigenous_ppl +
                           married_ppl + work_loc_foreign +
                           can_citizen_ppl + non_citizen_ppl,
                         data = data, family = quasipoisson())

# Summary of the model
summary(glm_quasi_poisson_model)
```

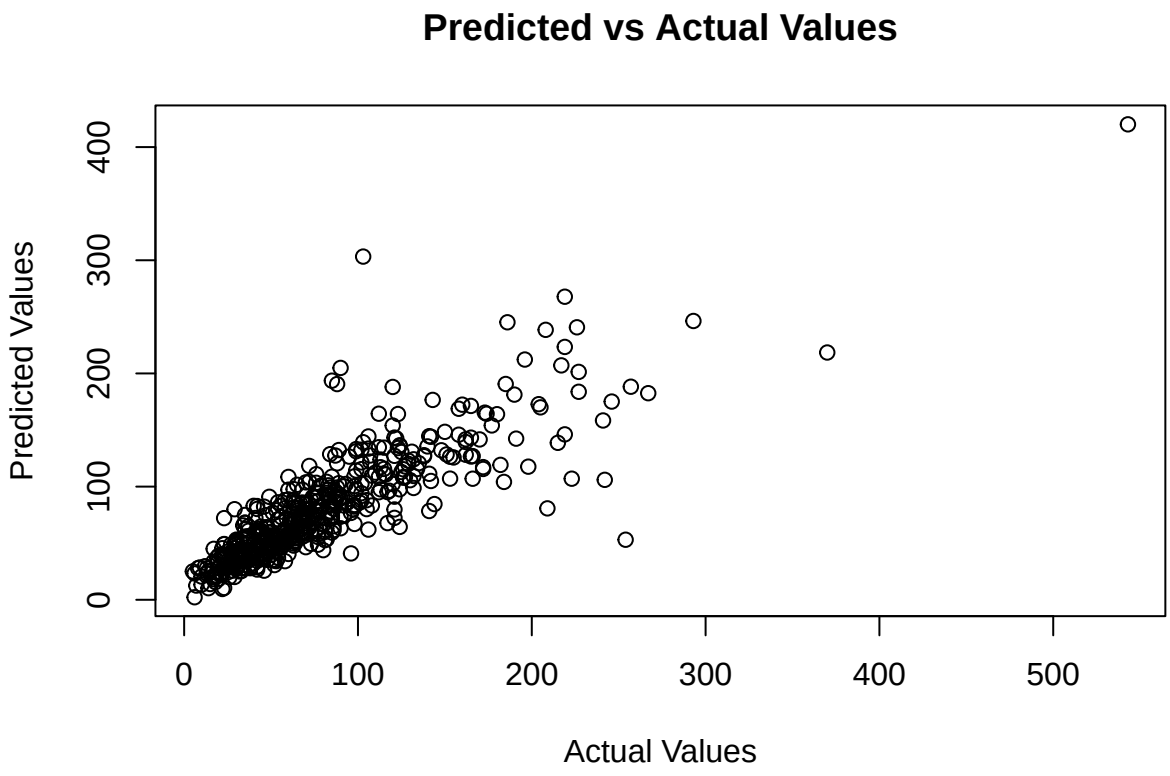
```
##
## Call:
## glm(formula = TotalEV ~ med_fulltime_income + income_100k_up +
##     med_total_household_income + income_40k + occup_cat_snr_mgmt +
##     occup_cat_busfin + occup_ind_realestate + work_loc_home +
##     work_loc_workplace + work_loc_notfixed + school_uni_degree +
##     school_hs + commute_start_noon + edu_field_science + edu_field_bus_admin +
##     edu_field_health + edu_field_pers_protect_transp + commute_transp_carpass +
##     commute_start_6am + commute_start_9am + condo + worktype_selfemp +
##     indigenous_ppl + married_ppl + work_loc_foreign + can_citizen_ppl +
##     non_citizen_ppl, family = quasipoisson(), data = data)
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   4.888e+00  9.691e-01   5.044 6.46e-07 ***
## med_fulltime_income           7.213e-06  4.726e-06   1.526 0.127576
## income_100k_up               -3.187e+00  1.373e+00  -2.321 0.020713 *
## med_total_household_income    4.399e-06  1.926e-06   2.284 0.022783 *
## income_40k                   -2.333e+00  3.261e+00  -0.715 0.474710
## occup_cat_snr_mgmt            2.099e+01  5.619e+00   3.736 0.000209 ***
## occup_cat_busfin             -2.078e+00  2.012e+00  -1.033 0.302148
## occup_ind_realestate          1.849e+01  5.722e+00   3.231 0.001318 **
## work_loc_home                 1.873e+00  1.152e+00   1.625 0.104785
## work_loc_workplace           -4.301e-01  1.207e+00  -0.356 0.721866
## work_loc_notfixed             7.035e+00  1.921e+00   3.662 0.000278 ***
## school_uni_degree             7.769e-01  1.094e+00   0.710 0.477814
## school_hs                    -2.577e+00  1.506e+00  -1.711 0.087717 .
```

```
## commute_start_noon             3.307e+00  2.549e+00   1.297 0.195199
## edu_field_science              8.140e+00  4.205e+00   1.936 0.053476 .
## edu_field_bus_admin            1.996e+00  1.672e+00   1.194 0.233075
## edu_field_health              -1.032e+00  1.945e+00  -0.531 0.595994
## edu_field_pers_protect_transp -3.207e+00  4.544e+00  -0.706 0.480688
## commute_transp_carpass         4.461e+00  4.062e+00   1.098 0.272679
## commute_start_6am             -3.451e-03  2.669e+00  -0.001 0.998969
## commute_start_9am              3.608e+00  2.839e+00   1.271 0.204340
## condo                          5.016e-01  1.975e-01   2.540 0.011387 *
## worktype_selfemp              -2.006e+00  1.414e+00  -1.419 0.156622
## indigenous_ppl                -3.972e+00  7.008e-01  -5.668 2.48e-08 ***
## married_ppl                    3.428e+00  6.720e-01   5.101 4.86e-07 ***
## work_loc_foreign               1.019e+01  4.725e+00   2.156 0.031552 *
## can_citizen_ppl               -3.470e+00  1.032e+00  -3.364 0.000829 ***
## non_citizen_ppl               -5.414e+00  1.167e+00  -4.641 4.47e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 9.37799)
##
##     Null deviance: 17307.1  on 513  degrees of freedom
## Residual deviance:  4127.4  on 486  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

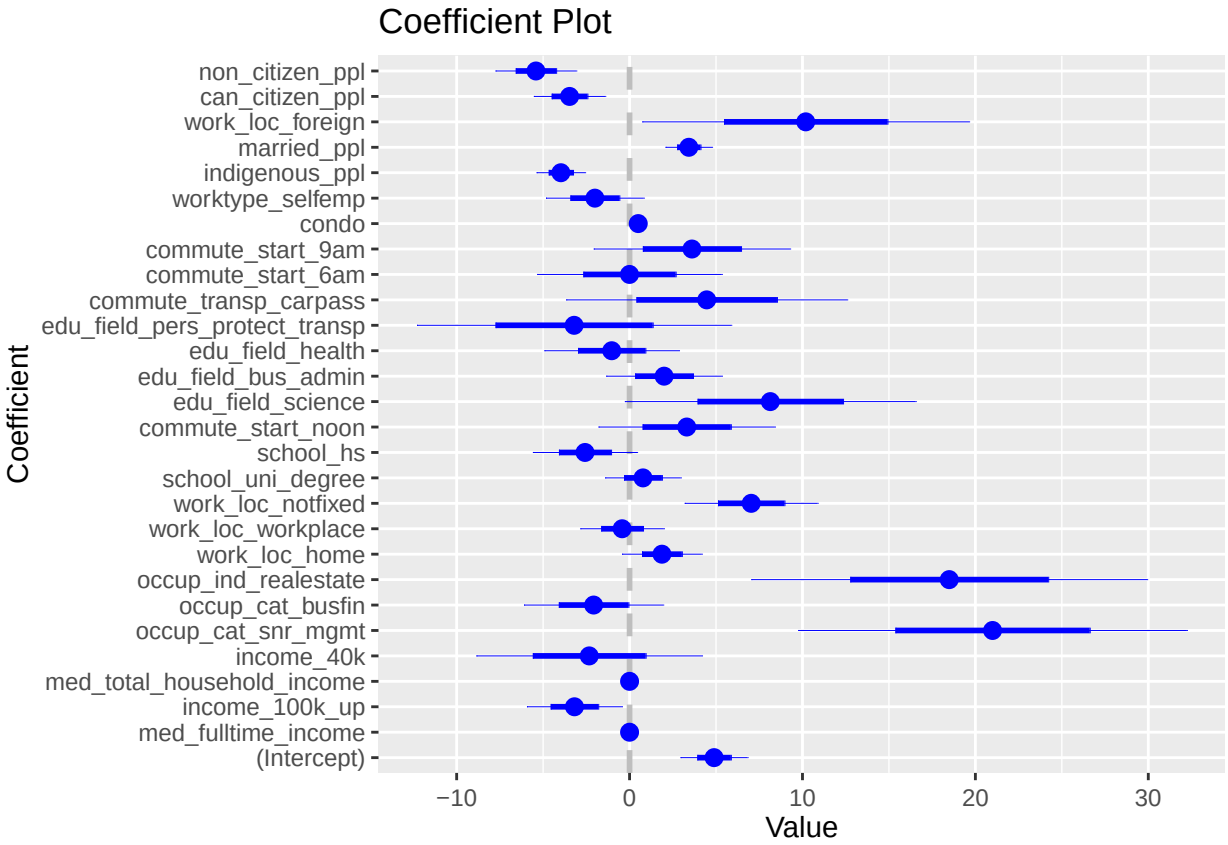
```r
#Residual plots
# Residuals vs Fitted Values
plot(glm_quasi_poisson_model$fitted.values, residuals(glm_quasi_poisson_model, type = "pearson"),
     xlab = "Fitted Values", ylab = "Pearson Residuals",
     main = "Residuals vs Fitted Values")
```




```r
# Predicted vs Actual Values
predicted_values <- predict(glm_quasi_poisson_model, type = "response")
actual_values <- data$TotalEV

plot(actual_values, predicted_values,
     xlab = "Actual Values", ylab = "Predicted Values",
     main = "Predicted vs Actual Values")
```




```r
# If necessary, install the coefplot package
if (!require("coefplot")) install.packages("coefplot")
library(coefplot)

# Plotting model coefficients
coefplot(glm_quasi_poisson_model)
```



The residual plots for the Quasi-Poisson model give us additional information on the model fit:

1. **Residuals vs Fitted Values Plot**: We still observe a pattern where residuals increase with the fitted values, indicating that the variance of the residuals is not constant and suggesting that the mean-variance relationship is not adequately captured by the Quasi-Poisson model. This pattern is a sign of overdispersion.

2. **Predicted vs Actual Values Plot**: There is a clear deviation from the line of equality (where predicted values equal actual values), especially for higher actual values. This indicates that the model is underpredicting for higher values of the response variable, which is another sign that the Poisson or Quasi-Poisson assumption might not be appropriate.

3. **Coefficient Plot**: The coefficient plot shows point estimates and confidence intervals for each of the model's coefficients. The widths of the confidence intervals vary, but some are quite wide, which suggests uncertainty in those estimates. The Quasi-Poisson model attempts to adjust the standard errors to account for overdispersion, but this does not necessarily improve the prediction accuracy.

Given these observations, the Quasi-Poisson model does not seem to be a good fit for the data. The patterns in the residuals and the inability of the model to predict higher values of the response variable accurately suggest that a model capable of handling overdispersion more explicitly could perform better.

A Negative Binomial GLM is a good alternative to try because it introduces an additional parameter to directly model overdispersion. The Negative Binomial model is particularly useful when the counts have an extra-Poisson variation, which seems to be the case here. It is often used for count data where the variance exceeds the mean.

In conclusion, considering the overdispersion evident in the residual plots, it would be advisable to fit a Negative Binomial GLM to the data and compare the model diagnostics, including the residual plots and

the AIC (if computable), with those from the Poisson and Quasi-Poisson models. The Negative Binomial GLM may provide a better fit and more reliable inference for our dataset.

```
# If necessary, install the MASS package
if (!require("MASS")) install.packages("MASS")
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:plotly':
##
##     select
```

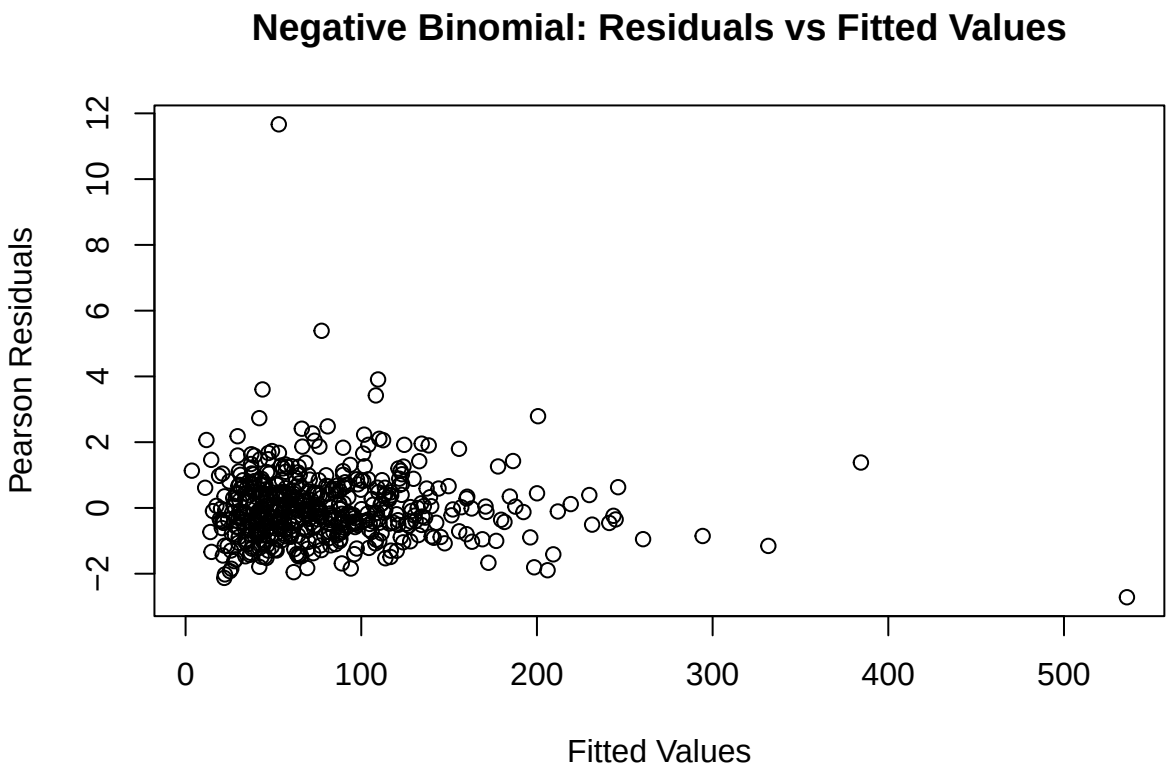
```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(MASS)

# Fit a GLM with a Negative Binomial distribution
glm_nb_model <- glm.nb(TotalEV ~ med_fulltime_income + income_100k_up +
                           med_total_household_income + income_40k +
                           occup_cat_snr_mgmt + occup_cat_busfin +
                           occup_ind_realestate + work_loc_home +
                           work_loc_workplace + work_loc_notfixed +
                           school_uni_degree + school_hs +
                           commute_start_noon + edu_field_science +
                           edu_field_bus_admin + edu_field_health +
                           edu_field_pers_protect_transp + commute_transp_carpass +
                           commute_start_6am + commute_start_9am +
                           condo + worktype_selfemp + indigenous_ppl +
                           married_ppl + work_loc_foreign +
                           can_citizen_ppl + non_citizen_ppl, data = data)

# Summary of the Negative Binomial model
summary(glm_nb_model)
```

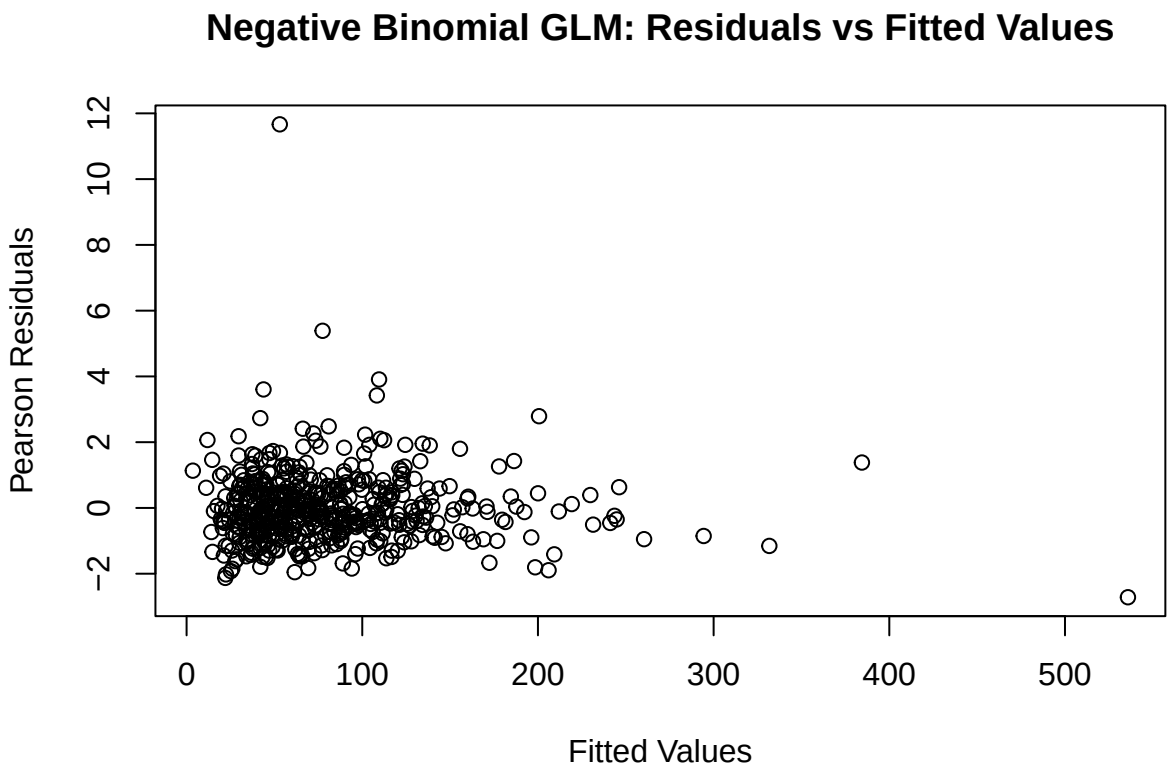
```
##
## Call:
## glm.nb(formula = TotalEV ~ med_fulltime_income + income_100k_up +
##     med_total_household_income + income_40k + occup_cat_snr_mgmt +
##     occup_cat_busfin + occup_ind_realestate + work_loc_home +
##     work_loc_workplace + work_loc_notfixed + school_uni_degree +
##     school_hs + commute_start_noon + edu_field_science + edu_field_bus_admin +
##     edu_field_health + edu_field_pers_protect_transp + commute_transp_carpass +
##     commute_start_6am + commute_start_9am + condo + worktype_selfemp +
##     indigenous_ppl + married_ppl + work_loc_foreign + can_citizen_ppl +
##     non_citizen_ppl, data = data, init.theta = 11.54673653, link = log)
##
## Coefficients:
```

```
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     5.134e+00  9.134e-01   5.621 1.90e-08 ***
## med_fulltime_income             3.955e-06  5.045e-06   0.784 0.433057
## income_100k_up                 -3.035e+00  1.416e+00  -2.143 0.032117 *
## med_total_household_income      8.416e-06  1.836e-06   4.584 4.56e-06 ***
## income_40k                      2.220e-01  2.928e+00   0.076 0.939568
## occup_cat_snr_mgmt              2.188e+01  6.056e+00   3.614 0.000302 ***
## occup_cat_busfin               -1.079e+00  2.013e+00  -0.536 0.591782
## occup_ind_realestate            1.835e+01  5.681e+00   3.230 0.001239 **
## work_loc_home                   1.599e+00  1.140e+00   1.402 0.160781
## work_loc_workplace             -1.997e+00  1.061e+00  -1.882 0.059814 .
## work_loc_notfixed               4.030e+00  1.791e+00   2.251 0.024412 *
## school_uni_degree               1.138e+00  1.051e+00   1.083 0.278818
## school_hs                      -8.146e-01  1.341e+00  -0.607 0.543537
## commute_start_noon              2.964e+00  2.286e+00   1.296 0.194852
## edu_field_science               1.279e+01  4.211e+00   3.037 0.002388 **
## edu_field_bus_admin            -1.642e-01  1.769e+00  -0.093 0.926059
## edu_field_health               -1.054e+00  1.856e+00  -0.568 0.570199
## edu_field_pers_protect_transp   1.251e+00  4.228e+00   0.296 0.767427
## commute_transp_carpass          5.574e+00  3.583e+00   1.555 0.119841
## commute_start_6am               2.418e+00  2.361e+00   1.024 0.305825
## commute_start_9am               5.942e+00  2.732e+00   2.175 0.029616 *
## condo                           6.940e-01  1.948e-01   3.563 0.000366 ***
## worktype_selfemp               -1.096e-01  1.321e+00  -0.083 0.933865
## indigenous_ppl                 -3.076e+00  4.298e-01  -7.157 8.22e-13 ***
## married_ppl                     2.683e+00  6.071e-01   4.419 9.90e-06 ***
## work_loc_foreign                4.228e+00  4.969e+00   0.851 0.394819
## can_citizen_ppl                -3.949e+00  9.611e-01  -4.109 3.97e-05 ***
## non_citizen_ppl                -5.847e+00  1.086e+00  -5.385 7.26e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(11.5467) family taken to be 1)
##
##     Null deviance: 2246.51  on 513  degrees of freedom
## Residual deviance:  521.22  on 486  degrees of freedom
## AIC: 4639.1
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  11.547
##           Std. Err.:  0.847
##
##  2 x log-likelihood:  -4581.095
```

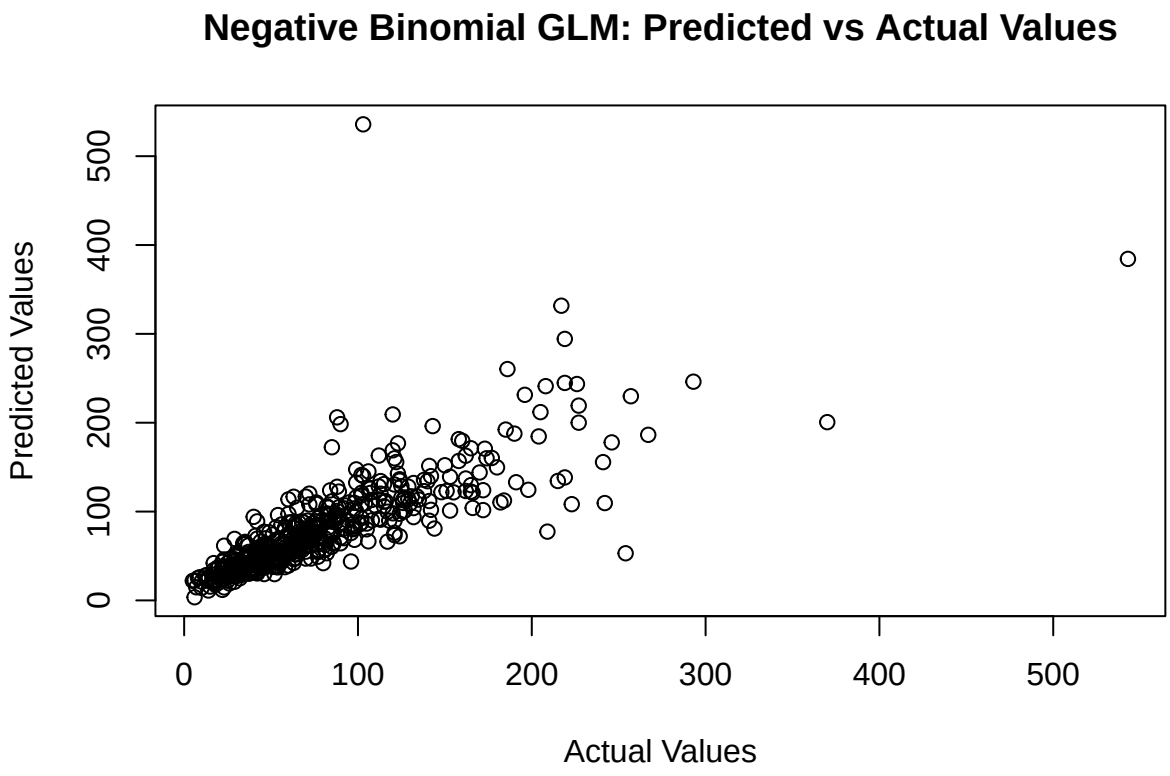
```r
# Residuals vs Fitted Values Plot for Negative Binomial
plot(glm_nb_model$fitted.values, residuals(glm_nb_model, type = "pearson"),
     xlab = "Fitted Values", ylab = "Pearson Residuals",
     main = "Negative Binomial: Residuals vs Fitted Values")
```



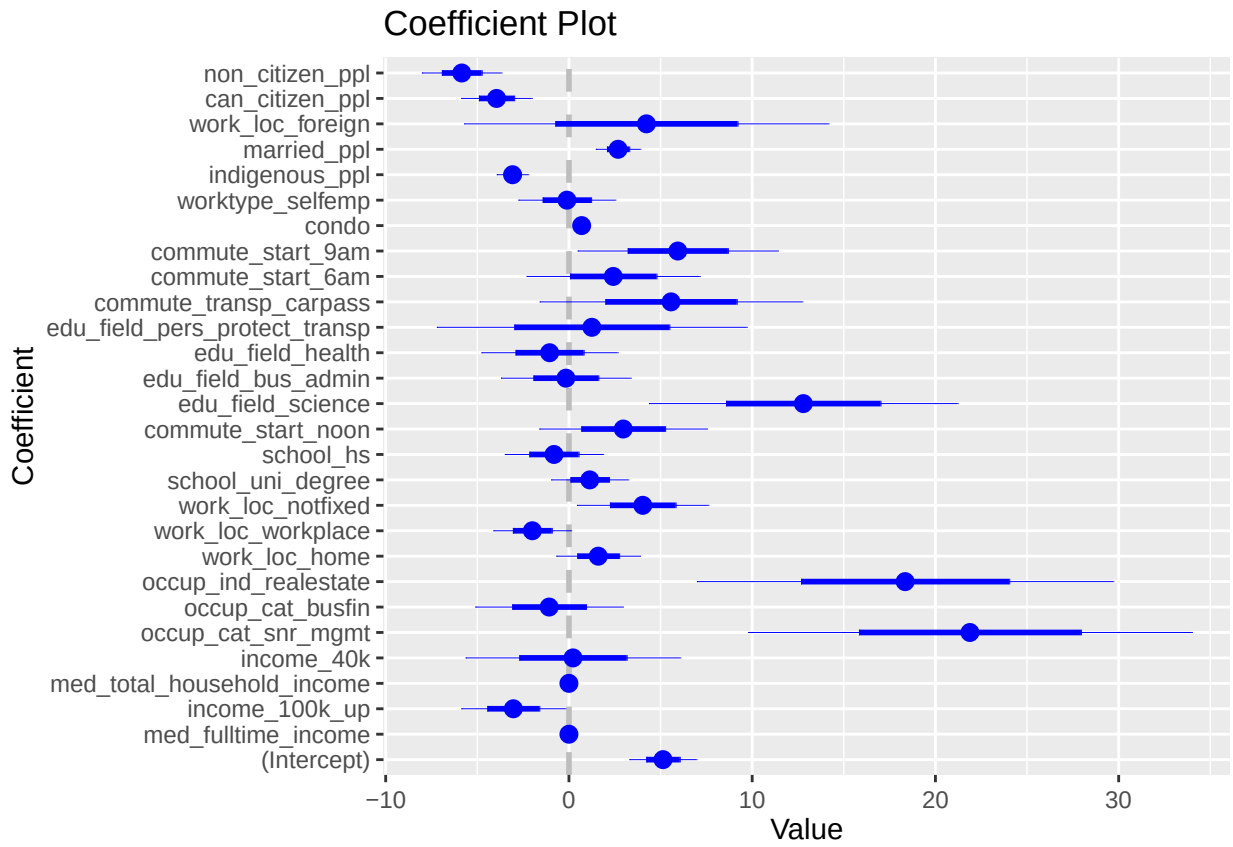

```r
# Plot 1: Residuals vs Fitted Values
plot(glm_nb_model$fitted.values, residuals(glm_nb_model, type = "pearson"),
     xlab = "Fitted Values", ylab = "Pearson Residuals",
     main = "Negative Binomial GLM: Residuals vs Fitted Values")
```




```r
# Plot 2: Predicted vs Actual Values
predicted_values <- predict(glm_nb_model, type = "response")
actual_values <- data$TotalEV

plot(actual_values, predicted_values,
     xlab = "Actual Values", ylab = "Predicted Values",
     main = "Negative Binomial GLM: Predicted vs Actual Values")
```




```
library(coefplot)

# Plotting model coefficients
coefplot(glm_nb_model)
```



The residual plots for the Negative Binomial GLM provide the following insights:

1. **Residuals vs Fitted Values Plot**: The plot shows an increasing trend in the magnitude of residuals as the fitted values increase. While the presence of overdispersion is much better accounted for in a Negative Binomial model than in a Poisson model, the pattern here indicates that there might still be some systematic variation not captured by the model. This could be due to several reasons, such as omitted variables that are important predictors, or it could suggest that the relationship between the predictors and the response variable might not be entirely linear.

2. **Predicted vs Actual Values Plot**: Similar to the Residuals vs Fitted plot, we see that the model tends to underpredict the actual values as they increase. There is a clear trend where the model does not capture the full range of the actual data, particularly for higher value responses. The points do not scatter randomly around the line of equality (where predicted values would equal actual values).

```
# # Function to calculate percentage change and IRR
# calculate_effect_size <- function(coefficient) {
#   percent_change <- (1 - exp(coefficient)) * 100
#   irr <- exp(coefficient)
#   return(c(percent_change = percent_change, irr = irr))
# }
#
# # Features based on the information provided (replace if necessary)
# features <- c("med_fulltime_income", "income_100k_up", "med_total_household_income",
#               "income_40k", "occup_cat_snr_mgmt", "occup_cat_busfin", #"occup_ind_realestate",
#               "work_loc_home", "work_loc_workplace", "work_loc_notfixed", #"school_uni_degree",
#               "school_hs", "commute_start_noon", "edu_field_science", #"edu_field_bus_admin",
```

```
#               "edu_field_pers_protect_transp", "commute_transp_carpass", #"commute_start_6am",
#               "commute_start_9am", "condo", "worktype_selfemp", "married_ppl",
#               "can_citizen_ppl", "non_citizen_ppl")
#
# # Assuming our model is stored in the object 'model'
# # Extract coefficients and standard errors
# coefficients <- coef(glm_nb_model)[names(coef(glm_nb_model)) %in% features]
# std_errors <- summary(glm_nb_model)$coefficients[ #,2][names(summary(glm_nb_model)$coefficients)[ ,1]
#
# features
#
# # Calculate effect size for each feature
# effect_sizes <- lapply(coefficients, calculate_effect_size)
# names(effect_sizes) <- features
#
# # Print results for significant features (adjust p-value threshold as needed)
# cat("Significant Features (p-value < 0.05):\n")
# significant_features <- features[summary(glm_nb_model)$coefficients[,4] < 0.05]
# if (length(significant_features) > 0) {
#   for (feature in significant_features) {
#     cat(paste0("* ", feature, ": \n"), sep="")
#     cat(paste0("  * Percent Change: ", round(effect_sizes[[feature]][1], 2), "%\n"), #sep="")
#     cat(paste0("  * Incidence Rate Ratio (IRR): ", round(effect_sizes[[feature]][2], 2), "\n"), sep="
#
#   }
# } else {
#   cat("No significant features found.\n")
# }
#
# # Identify most influential features based on absolute percentage change
# most_influential <- features[abs(effect_sizes) %%=%% "percent_change" == #max(abs(unlist(effect_sizes
#
# cat("\nMost Influential Features (by absolute percentage change):\n")
# cat(paste(most_influential, collapse = ", "))
#
#
# # Check for missing coefficients
# summary(glm_nb_model)
#
# # Remove features with missing coefficients (optional)
# significant_features <- significant_features[!(is.na(coef(glm_nb_model)[significant_features]))]
#
# if (length(significant_features) > 0) {
#   for (feature in significant_features) {
#     # Check for non-numeric values (e.g., NA)
#     if (!is.numeric(effect_sizes[[feature]][1])) {
#       cat("WARNING: Missing effect size for", feature, "\n")
#       next  # Skip to the next iteration
#     }
#     cat(paste0("* ", feature, ": \n"), sep="")
#     cat(paste0("  * Percent Change: ", round(effect_sizes[[feature]][1], 2), "%\n"), sep="")
#     cat(paste0("  * Incidence Rate Ratio (IRR): ", round(effect_sizes[[feature]][2], 2), "\n"), sep="
#   }
```

```r
# } else {
#   cat("No significant features found.\n")
# }


model_summary <- summary(glm_nb_model)
std_coefficients <- model_summary$coefficients[, "z value"]


# Extract p-values
p_values <- model_summary$coefficients[, "Pr(>|z|)"]

# Filter for significant features
significant_std_coefficients <- std_coefficients[p_values < 0.05]

# Get the names of the top 5 significant features by absolute z-value
top_significant_features <- sort(abs(significant_std_coefficients), decreasing = TRUE)[1:5]
top_significant_feature_names <- names(top_significant_features)

# Print the top 5 significant features and their standardized coefficients
cat("Top 5 most influential significant features based on the absolute z values:\n")
```

```
## Top 5 most influential significant features based on the absolute z values:
```

```r
for (feature_name in top_significant_feature_names) {
  cat(paste0(feature_name, ": ", round(significant_std_coefficients[feature_name], 2)), "\n")
}
```

```
## indigenous_ppl: -7.16
## (Intercept): 5.62
## non_citizen_ppl: -5.38
## med_total_household_income: 4.58
## married_ppl: 4.42
```

```r
# Get model's coefficients
coefficients <- coef(glm_nb_model)

# Calculate Incidence Rate Ratios (IRR) by exponentiating the coefficients
irr <- exp(coefficients)

# Sort features by the magnitude of their IRR (from most to least influential) and get top 5
top_features <- sort(irr, decreasing = TRUE)[1:5]

# Print the top 5 features and their IRR
cat("Top 5 features by their influence on TotalEV (Incidence Rate Ratios):\n")
```

```
## Top 5 features by their influence on TotalEV (Incidence Rate Ratios):
```

```r
print(top_features)
```

```
##     occup_cat_snr_mgmt   occup_ind_realestate      edu_field_science
```

```
##           3.193502e+09           9.293642e+07           3.589418e+05
##      commute_start_9am commute_transp_carpass
##           3.806251e+02           2.634068e+02
```

Our goal in the first research question is to understand the strength and direction of the effect of predictors on the response variable, IRR is more appropriate because it gives a multiplicative effect size.

We have used the IRR from the GLM Model with a negative binomial distribution to understand and communicate the effect size and direction of the predictors here. The occupation, commute start time and the mode of commute and the education field play an important role in the Total Electric Vehicle count from our findings.

Research Question 2

```
set.seed(1234)
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.3.3
```

```
## Linking to GEOS 3.11.2, GDAL 3.8.2, PROJ 9.3.1; sf_use_s2() is TRUE
```

```
library(ggplot2)
library(AER)
```

```
## Warning: package 'AER' was built under R version 4.3.3
```

```
## Loading required package: car
```

```
## Warning: package 'car' was built under R version 4.3.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.3.3
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## Loading required package: lmtest
```

```
## Warning: package 'lmtest' was built under R version 4.3.3
```

```
## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.3.3

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.3.3

## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster
```

```
library(MASS)
library(sfdep)
```

```
## Warning: package 'sfdep' was built under R version 4.3.3

##
## Attaching package: 'sfdep'

## The following object is masked from 'package:car':
##
##     ellipse
```

```
library(spdep)
```

```
## Warning: package 'spdep' was built under R version 4.3.3

## Loading required package: spData

## Warning: package 'spData' was built under R version 4.3.3

## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

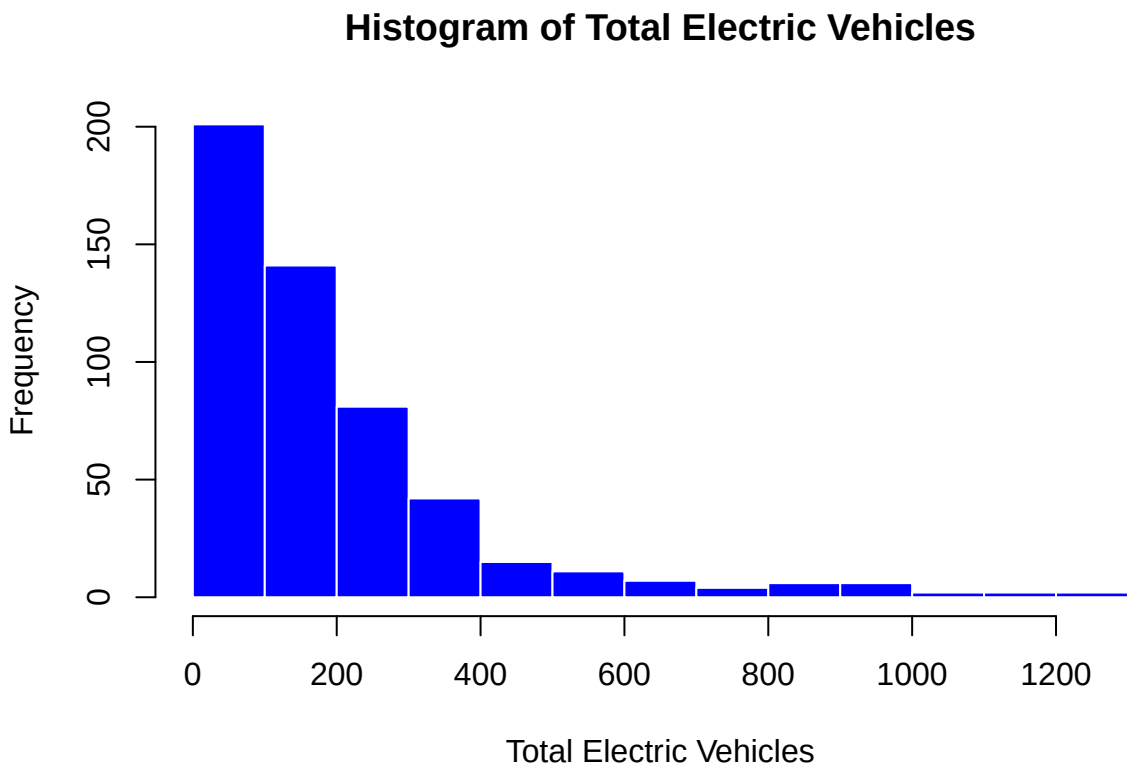
```r
library(tidyverse)
library(dplyr)
```

```r
# import shape file for Canada
canada <- st_read("lfsa000a21a_e.shp")
```

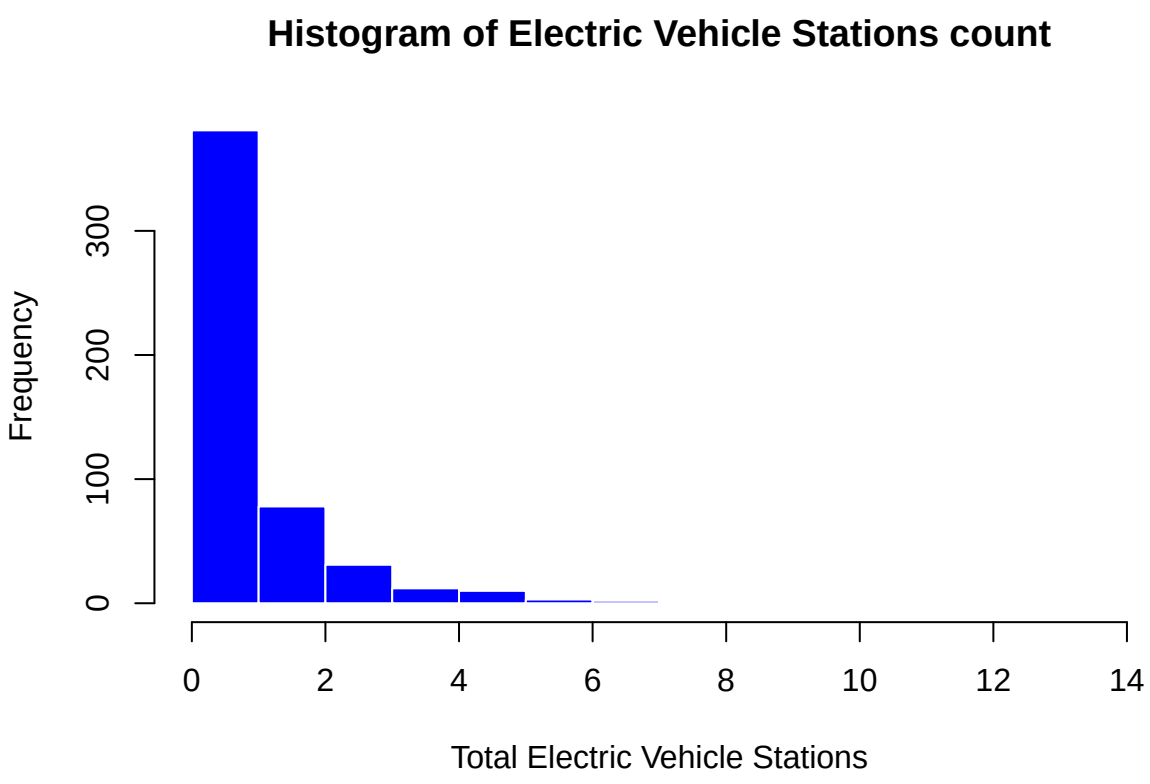
```
## Reading layer 'lfsa000a21a_e' from data source
##   'C:\Users\Allotei\Desktop\dalhousie\23-24 Winter\Data Analysis\Project\lfsa000a21a_e.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 1643 features and 5 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 3658201 ymin: 658873 xmax: 9019157 ymax: 6083005
## Projected CRS: NAD83 / Statistics Canada Lambert
```

```r
# Subset to get only Ontario
ontario <- canada[canada$PRNAME == "Ontario", ]

# get charging stations in ontario
charging_stations <- read.csv('fuel_stations.csv')

# charging stations as coordinates
charging_stations_coords <- st_as_sf(charging_stations, coords = c("Longitude", "Latitude"), crs = 4326)

# load data with ev counts and charging stations
data_with_ev_counts_stations <- read.csv('electric_vehicles.csv')

# data new
data_new <- read.csv('data_new.csv')

# subset with only station count and fsa
data_new <- data_new[, c("FSA", "CountOfEVStations")]

# ontario data
ontario_data <- merge(ontario, data_with_ev_counts_stations, by.x = "CFSAUID", by.y = "FSA", all.x = TRU
ontario_data <- merge(ontario_data, data_new, by.x = "CFSAUID", by.y = "FSA", all.x = TRUE)

# Replacing NA with 0
ontario_data$TotalEV[is.na(ontario_data$TotalEV)] <- 0
ontario_data$CountOfEVStations[is.na(ontario_data$CountOfEVStations)] <- 0
```

```r
# histogram of counts of ev vehicles
hist(ontario_data$TotalEV,
     main = "Histogram of Total Electric Vehicles",
     xlab = "Total Electric Vehicles",
     col = "blue",
     border = "white")
```




```r
# ev count data seems poisson distributed

hist(ontario_data$CountOfEVStations,
     main = "Histogram of Electric Vehicle Stations count",
     xlab = "Total Electric Vehicle Stations",
     col = "blue",
     border = "white")
```




```
# ev station count data also seems poisson distributed

# checking for overdispersion
mean(ontario_data$TotalEV)
```

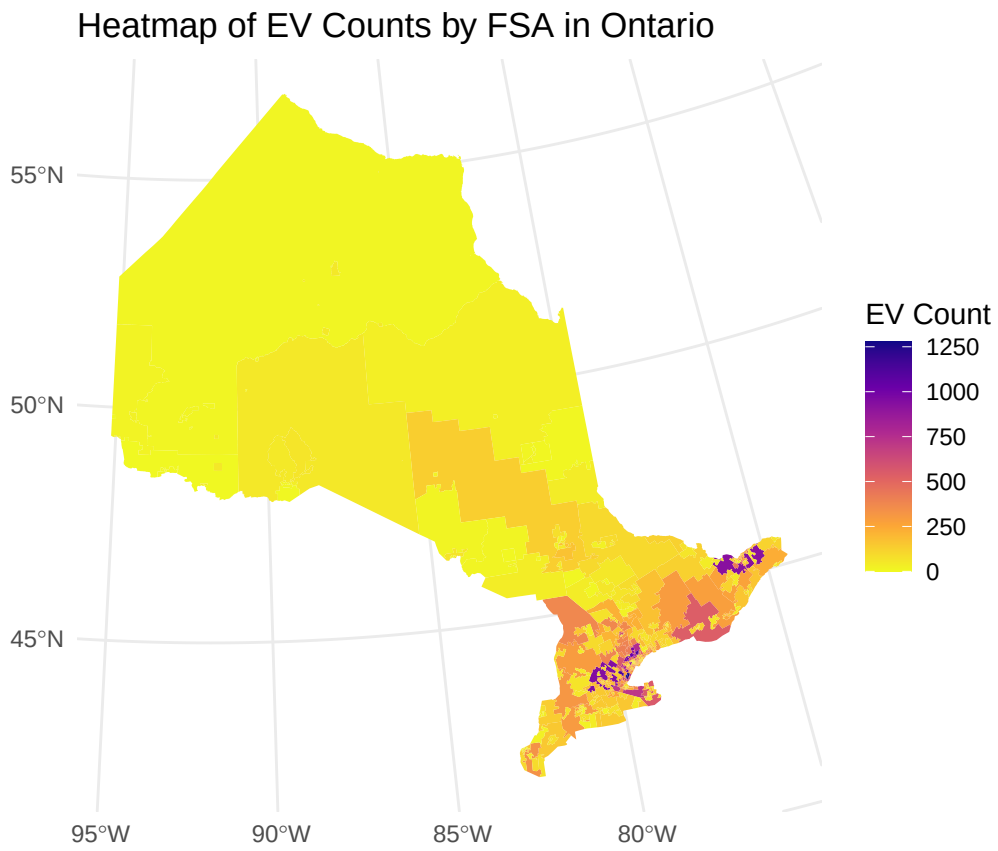
```
## [1] 199.9519
```

```
var(ontario_data$TotalEV)
```

```
## [1] 43389.36
```

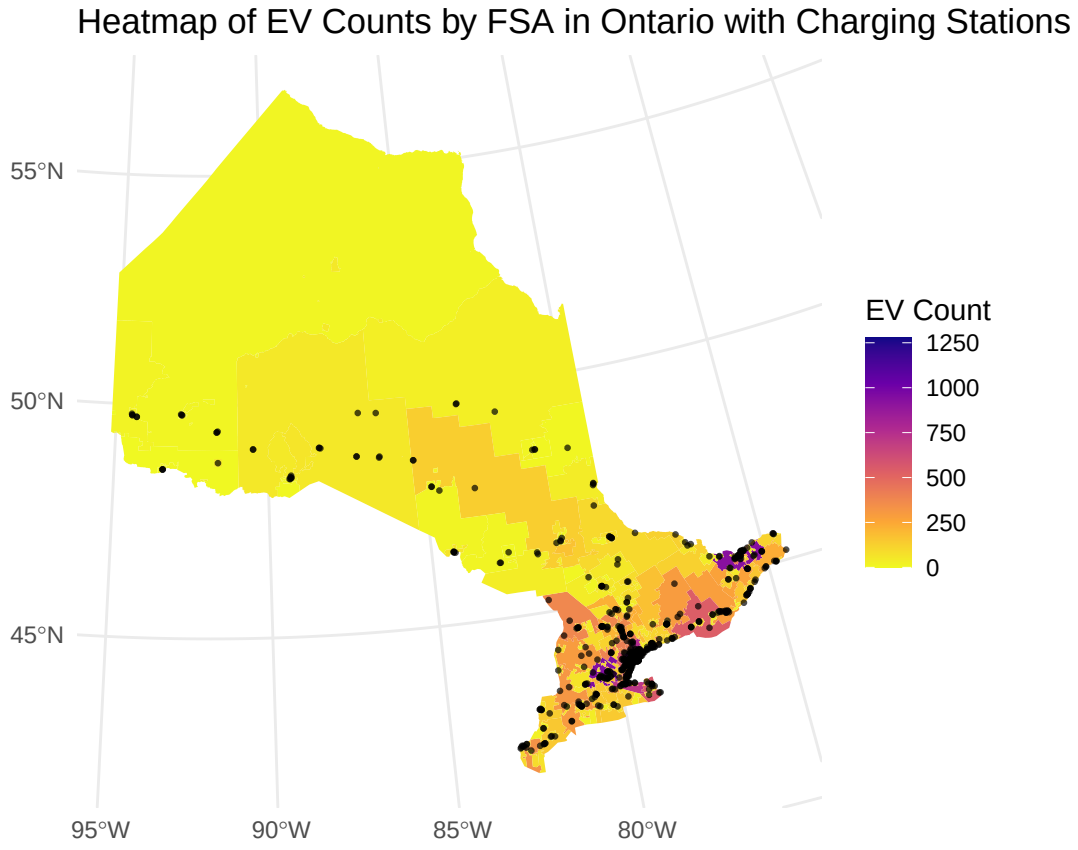
```
# variance far exceeds mean. suggests overdispersion. thus may not be poisson distributed
# as assumed
```

```
ggplot(data = ontario_data) +
  geom_sf(aes(fill = TotalEV), color = NA) + # Fill based on EV_Count
  scale_fill_viridis_c(option = "plasma", direction = -1, name = "EV Count") + # Use a color scale
  ggtitle("Heatmap of EV Counts by FSA in Ontario") +
  theme_minimal() +
  coord_sf() # Use coord_sf to maintain aspect ratio
```




```
p = ggplot(data = ontario_data) +
  geom_sf(aes(fill = TotalEV), color = NA) + # Fill based on EV_Count
  scale_fill_viridis_c(option = "plasma", direction = -1, name = "EV Count") + # Use a color scale
  ggtitle("Heatmap of EV Counts by FSA in Ontario with Charging Stations") +
  theme_minimal() +
  coord_sf() # Use coord_sf to maintain aspect ratio

p + geom_sf(data = charging_stations_coords, inherit.aes = FALSE, color = "black", size =0.5, alpha = 0
```

```
## Coordinate system already present. Adding new coordinate system, which will
## replace the existing one.
```

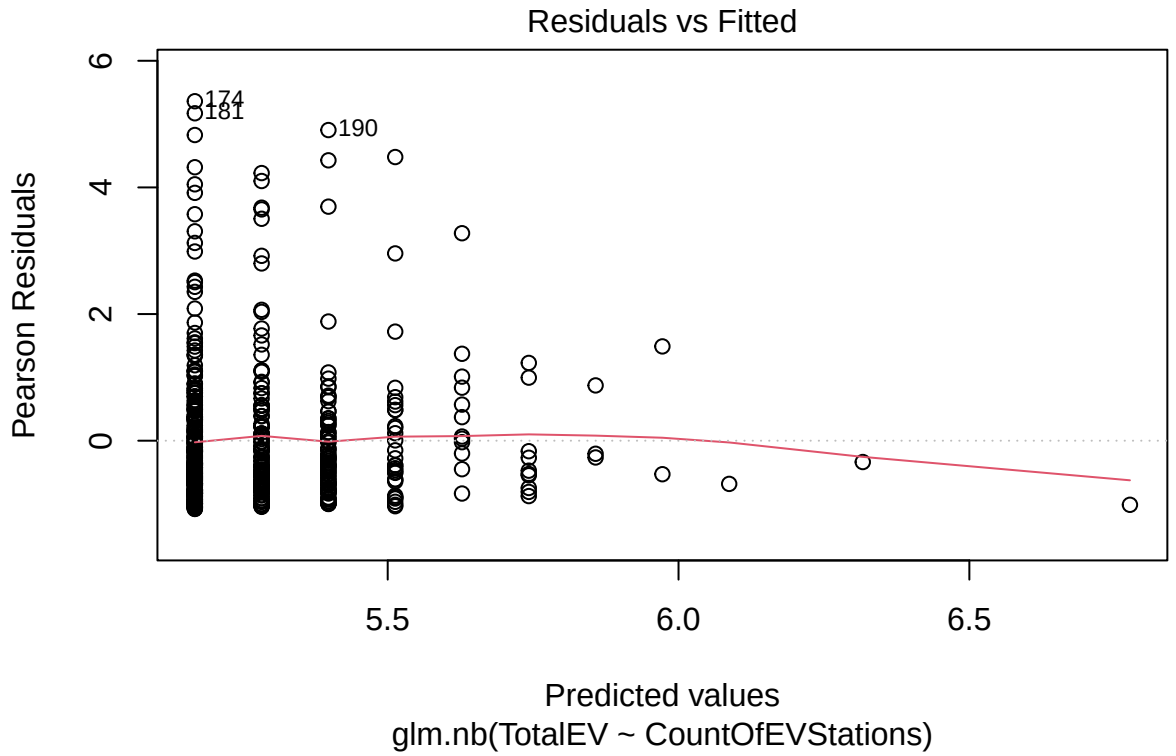


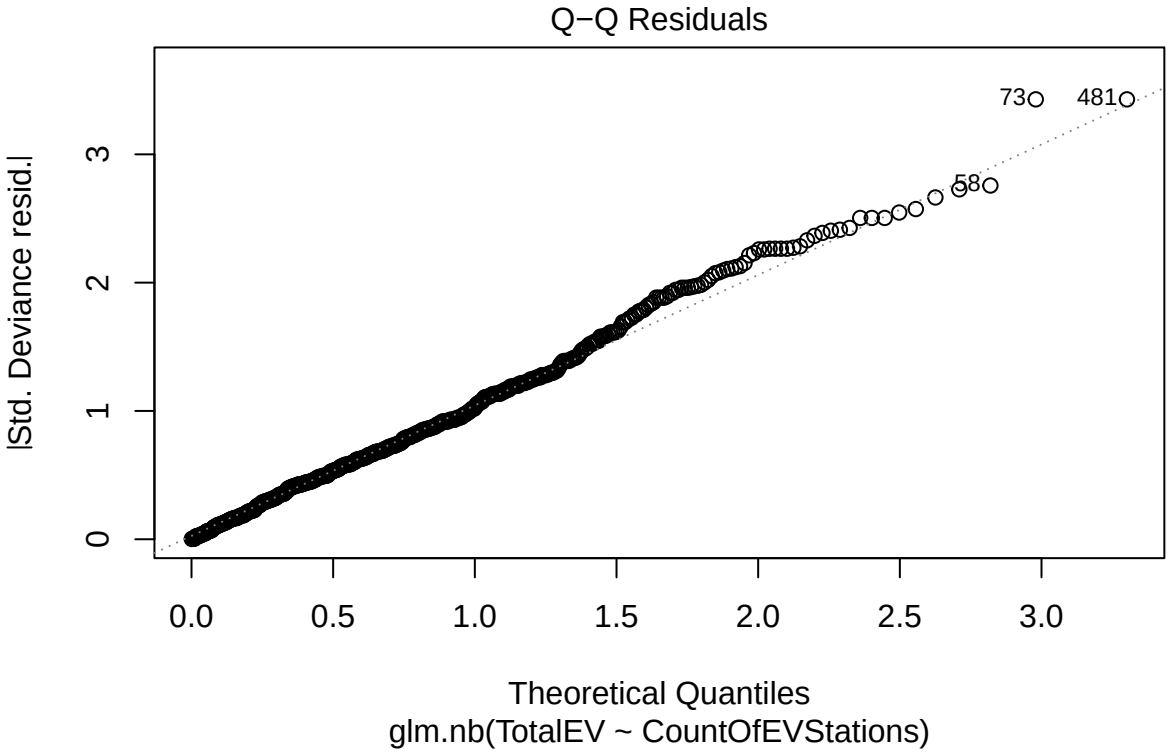

```
# trying the Poisson either way
M1<-glm(TotalEV~CountOfEVStations,family=poisson,data=ontario_data)
#summary(M1)
#plot(M1)

# Perform the dispersion test
#disp_test <- dispersiontest(M1)

# Print the test result
#print(disp_test)

# after dispersion test on poisson model reveals overdispersion in the model

# trying a negative binomial model instead
M2 <- glm.nb(TotalEV~CountOfEVStations, data=ontario_data)
summary(M2)
```

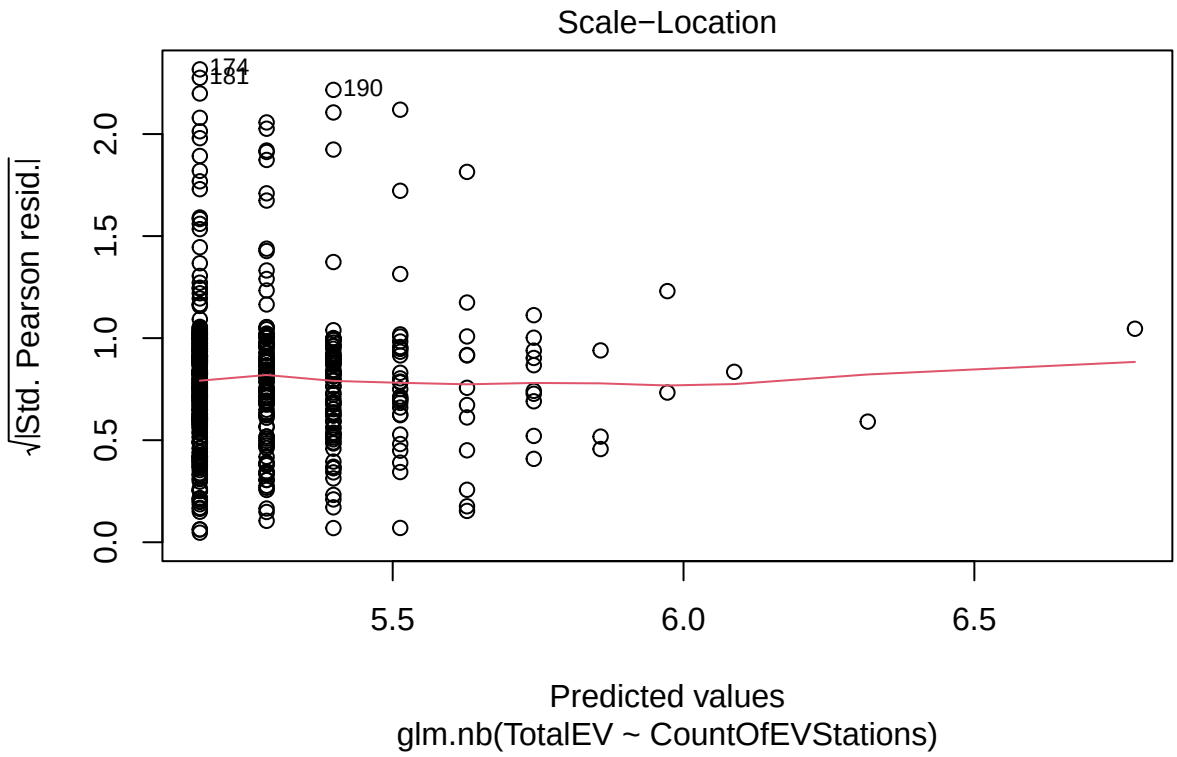
```
##
## Call:
## glm.nb(formula = TotalEV ~ CountOfEVStations, data = ontario_data,
##     init.theta = 1.167542588, link = log)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        5.16845    0.04885 105.806  < 2e-16 ***
## CountOfEVStations  0.11483    0.02692   4.266 1.99e-05 ***
```

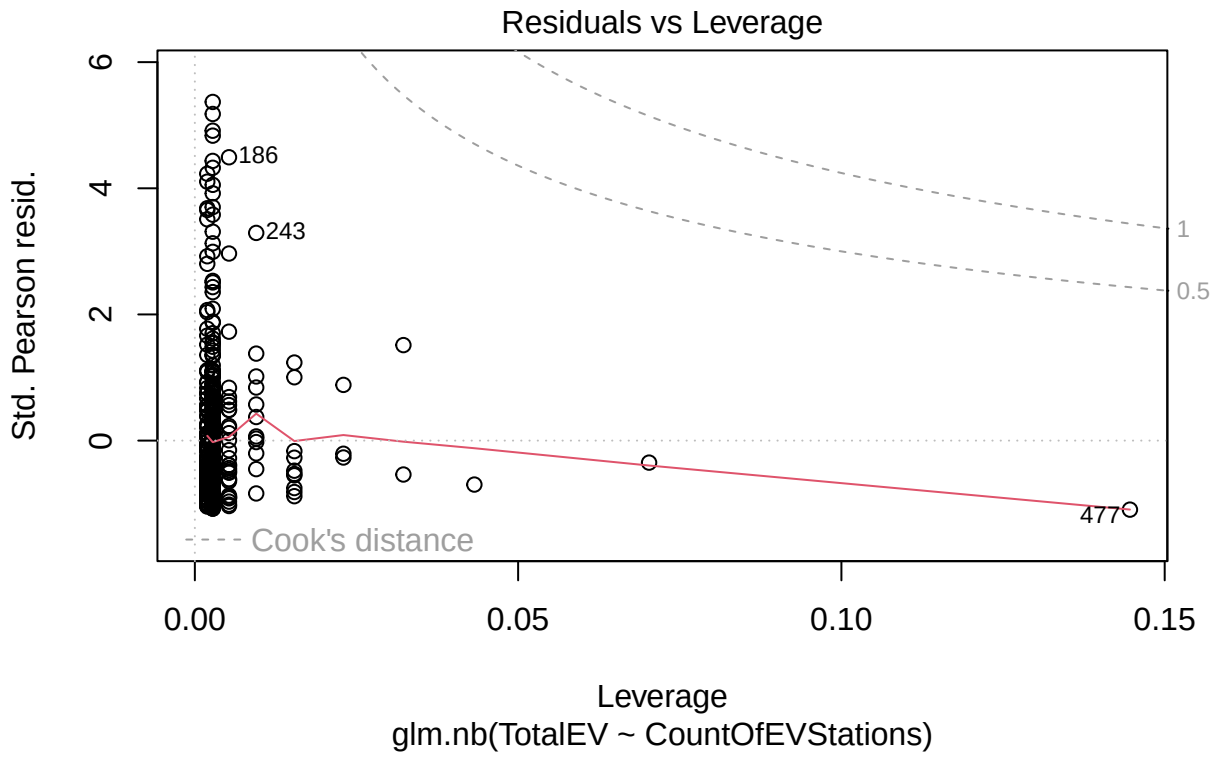
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.1675) family taken to be 1)
##
##     Null deviance: 608.11  on 519  degrees of freedom
## Residual deviance: 590.50  on 518  degrees of freedom
## AIC: 6536.2
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.1675
##           Std. Err.:  0.0658
##
##  2 x log-likelihood:  -6530.2270
```

```
plot(M2)
```




Q−Q Residuals
|Std. Deviance resid.|
0
1
2
3
0.0
0.5
1.0
1.5
2.0
2.5
3.0
73
481
58
Theoretical Quantiles
glm.nb(TotalEV ~ CountOfEVStations)

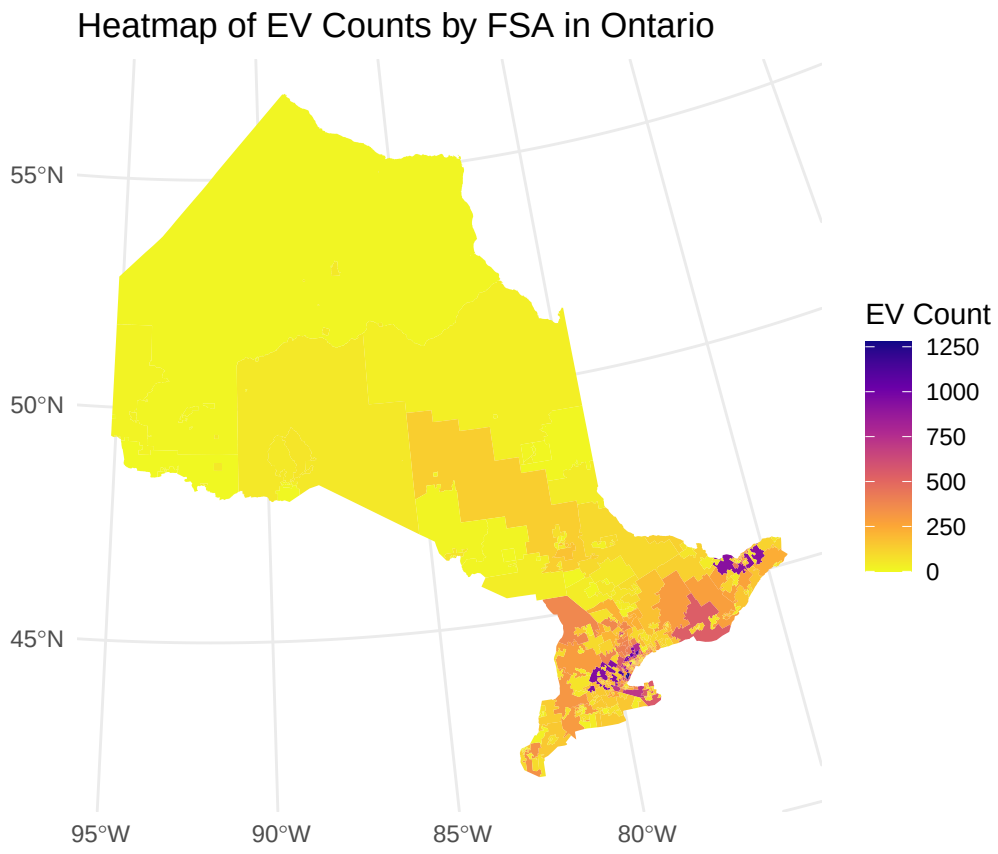

Scale−Location
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
2.0
174
181
190
5.5
6.0
6.5
Predicted values
glm.nb(TotalEV ~ CountOfEVStations)




```
llneg <- logLik(M2)
llp <- logLik(M1)
llstatistic <- 2 * (llneg - llp)
p_value <- pchisq(llstatistic, df=1, lower.tail=FALSE)
p_value
```

```
## 'log Lik.' 0 (df=3)
```

```
# visualize ev station count across fsa
ggplot(data = ontario_data) +
  geom_sf(aes(fill = TotalEV), color = NA) + # Fill based on EV_Count
  scale_fill_viridis_c(option = "plasma", direction = -1, name = "EV Count") + # Use a color scale
  ggtitle("Heatmap of EV Counts by FSA in Ontario") +
  theme_minimal() +
  coord_sf() # Use coord_sf to maintain aspect ratio
```




```
# create a neighbor list based on queen contiguity
list_nb <- poly2nb(ontario_data, queen = TRUE)
```

```
empty_nb <- which(card(list_nb) == 0)
empty_nb
```

```
## integer(0)
```

```
# identify neighbors with queen contiguity
ev_nb <- poly2nb(ontario_data, queen = TRUE)

ev_w_binary <- nb2listw(ev_nb, style="B")

ev_lag <- lag.listw(ev_w_binary, ontario_data$TotalEV)
```

```
globalG.test(ontario_data$TotalEV, ev_w_binary)
```

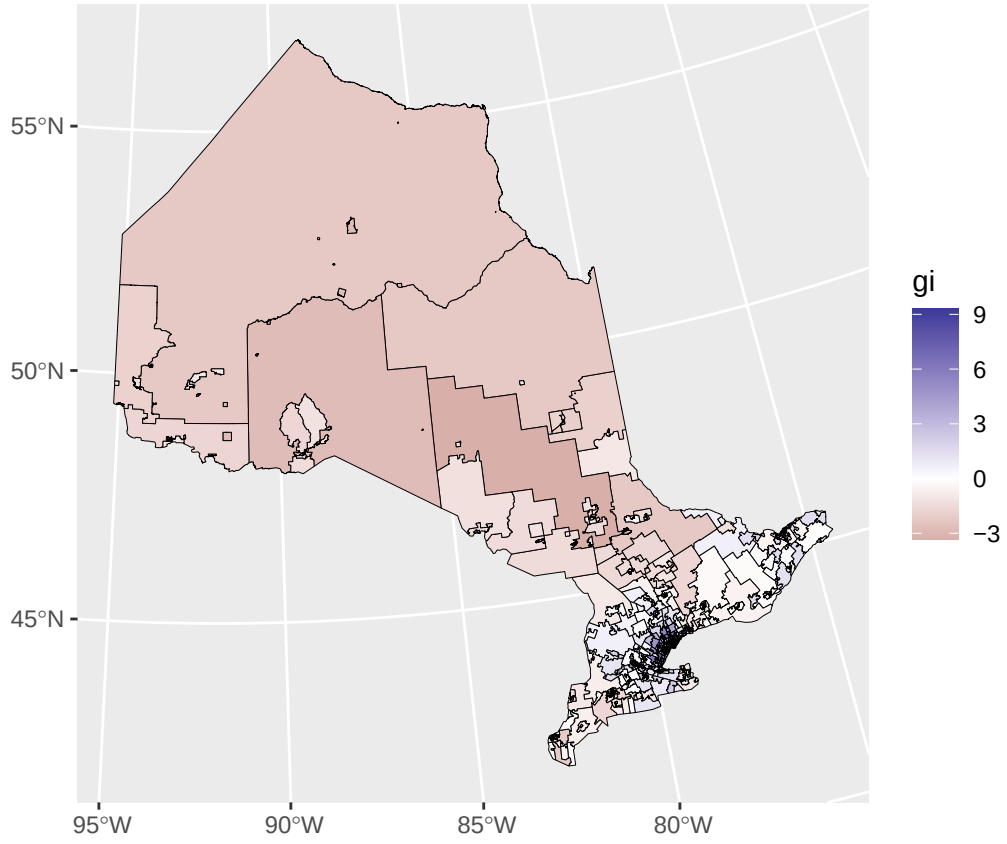
```
##
##  Getis-Ord global G statistic
##
## data:  ontario_data$TotalEV
## weights: ev_w_binary
##
## standard deviate = 15.847, p-value < 2.2e-16
```

```
## alternative hypothesis: greater
## sample estimates:
## Global G statistic        Expectation           Variance
##       2.083694e-02       1.080480e-02       4.007627e-07
```

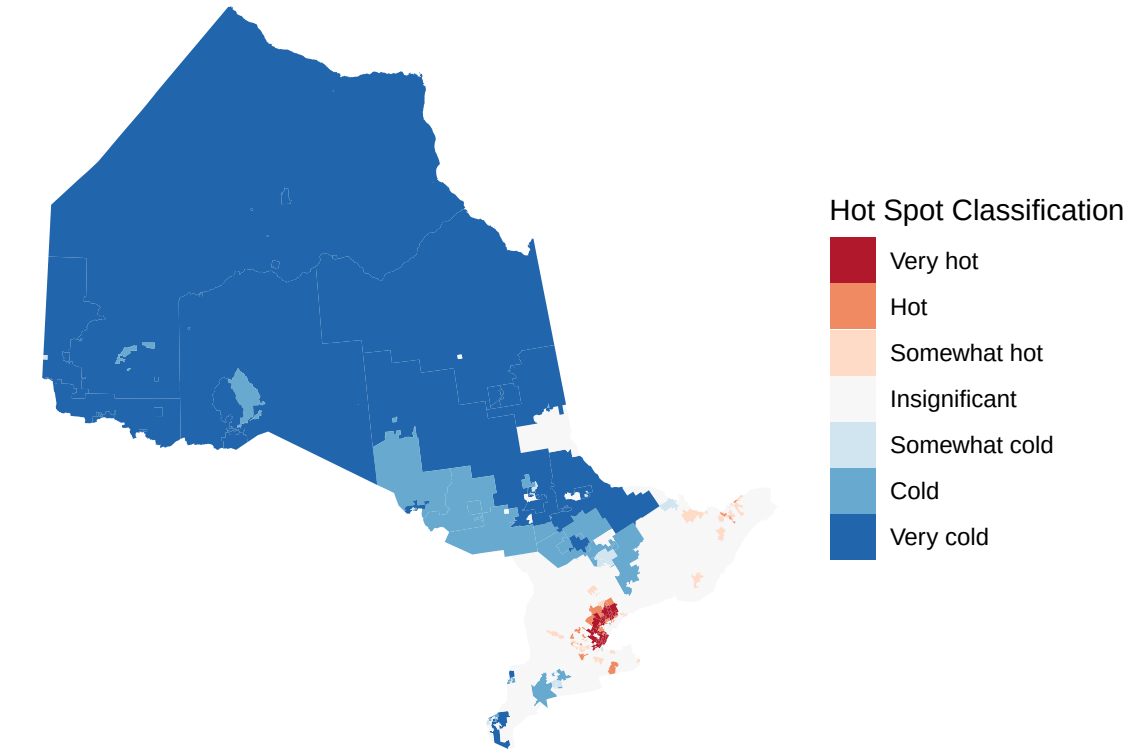
```
ev_nbs <- ontario_data |>
  mutate(
    nb = st_contiguity(geometry),
    wt = st_weights(nb),
    ev_lag = st_lag(TotalEV, nb, wt)
  )
```

```
ev_hot_spots <- ev_nbs |>
  mutate(
    Gi = local_g_perm(TotalEV, nb, wt, nsim=999)
  ) |>
  unnest(Gi)
```

```
ev_hot_spots |>
  ggplot((aes(fill = gi))) +
  geom_sf(color="black", lwd=0.15) +
  scale_fill_gradient2()
```



```r
ev_hot_spots |>
  # with the columns 'gi' and 'p_folded_sim"
  # 'p_folded_sim' is the p-value of a folded permutation test
  dplyr::select(gi, p_folded_sim) |>
  mutate(
    # Add a new column called "classification"
    classification = case_when(
      # Classify based on the following criteria:
      gi > 0 & p_folded_sim <= 0.01 ~ "Very hot",
      gi > 0 & p_folded_sim <= 0.05 ~ "Hot",
      gi > 0 & p_folded_sim <= 0.1 ~ "Somewhat hot",
      gi < 0 & p_folded_sim <= 0.01 ~ "Very cold",
      gi < 0 & p_folded_sim <= 0.05 ~ "Cold",
      gi < 0 & p_folded_sim <= 0.1 ~ "Somewhat cold",
      TRUE ~ "Insignificant"
    ),
    # Convert 'classification' into a factor for easier plotting
    classification = factor(
      classification,
      levels = c("Very hot", "Hot", "Somewhat hot",
                 "Insignificant",
                 "Somewhat cold", "Cold", "Very cold")
    )
  ) |>
  # Visualize the results with ggplot2
  ggplot(aes(fill = classification)) +
  geom_sf(color = NA, lwd = 0.1) +
  scale_fill_brewer(type = "div", palette = 5) +
  theme_void() +
  labs(
    fill = "Hot Spot Classification",
    title = "EV Counts Hot Spots in Ontario"
  )
```



Research Question 3

```
library(tidyverse)
library(readr)
library(caret)
library(randomForest)
library(rpart)

# Load the dataset
data_ev <- read_csv("./data_ev.csv", show_col_types = FALSE)
```

```
## New names:
## * `` -> `...1`
```

```
data_ev <- data_ev[,-1] # Removes the first column

# Check the first few rows of the dataset
head(data_ev)
```

```
## # A tibble: 6 x 29
##   FSA   med_fulltime_income income_100k_up med_total_household_income income_40k
##   <chr>               <dbl>          <dbl>                      <dbl>      <dbl>
## 1 K0A                 74500          0.127                     115000     0.0789
```

```
## 2 K0B                 56800         0.0635                      79000     0.0980
## 3 K0C                 58800         0.0664                      84000     0.0961
## 4 K0E                 58000         0.0636                      83000     0.103
## 5 K0G                 64500         0.0945                      94000     0.0926
## 6 K0H                 64000         0.0829                      92000     0.0903
## # i 24 more variables: occup_cat_snr_mgmt <dbl>, occup_cat_busfin <dbl>,
## #   occup_ind_realestate <dbl>, work_loc_home <dbl>, work_loc_workplace <dbl>,
## #   work_loc_notfixed <dbl>, school_uni_degree <dbl>, school_hs <dbl>,
## #   commute_start_noon <dbl>, edu_field_science <dbl>,
## #   edu_field_bus_admin <dbl>, edu_field_health <dbl>,
## #   edu_field_pers_protect_transp <dbl>, commute_transp_carpass <dbl>,
## #   commute_start_6am <dbl>, commute_start_9am <dbl>, condo <dbl>, ...
```

```
# Summarize the dataset
summary(data_ev)
```

```
##      FSA            med_fulltime_income income_100k_up
##  Length:518         Min.   : 44400      Min.   :0.01273
##  Class :character   1st Qu.: 58800      1st Qu.:0.06148
##  Mode  :character   Median : 65500      Median :0.08621
##                     Mean   : 68139      Mean   :0.10007
##                     3rd Qu.: 75500      3rd Qu.:0.12880
##                     Max.   :118000      Max.   :0.28238
##                     NA's   :4           NA's   :3
##  med_total_household_income   income_40k      occup_cat_snr_mgmt
##  Min.   : 45200             Min.   :0.04717   Min.   :0.000000
##  1st Qu.: 76750             1st Qu.:0.07342   1st Qu.:0.003372
##  Median : 90000             Median :0.08500   Median :0.005380
##  Mean   : 94219             Mean   :0.08443   Mean   :0.007227
##  3rd Qu.:110000             3rd Qu.:0.09664   3rd Qu.:0.008901
##  Max.   :198000             Max.   :0.11766   Max.   :0.045506
##  NA's   :3                  NA's   :3         NA's   :3
##  occup_cat_busfin  occup_ind_realestate work_loc_home     work_loc_workplace
##  Min.   :0.03298   Min.   :0.000000     Min.   :0.02332   Min.   :0.06734
##  1st Qu.:0.06893   1st Qu.:0.006916     1st Qu.:0.07926   1st Qu.:0.24020
##  Median :0.08676   Median :0.009223     Median :0.11263   Median :0.26862
##  Mean   :0.09126   Mean   :0.010450     Mean   :0.13247   Mean   :0.26775
##  3rd Qu.:0.10934   3rd Qu.:0.012723     3rd Qu.:0.17023   3rd Qu.:0.29800
##  Max.   :0.27591   Max.   :0.033793     Max.   :0.48062   Max.   :0.42754
##  NA's   :3         NA's   :3            NA's   :3         NA's   :3
##  work_loc_notfixed school_uni_degree   school_hs      commute_start_noon
##  Min.   :0.00000   Min.   :0.04198   Min.   :0.0155   Min.   :0.00000
##  1st Qu.:0.04328   1st Qu.:0.10869   1st Qu.:0.1029   1st Qu.:0.03979
##  Median :0.05146   Median :0.17016   Median :0.1272   Median :0.05140
##  Mean   :0.05273   Mean   :0.18947   Mean   :0.1251   Mean   :0.05116
##  3rd Qu.:0.06135   3rd Qu.:0.24548   3rd Qu.:0.1497   3rd Qu.:0.06209
##  Max.   :0.13049   Max.   :0.59883   Max.   :0.1989   Max.   :0.10359
##  NA's   :3         NA's   :3         NA's   :3        NA's   :3
##  edu_field_science  edu_field_bus_admin edu_field_health
##  Min.   :0.002772   Min.   :0.02525     Min.   :0.00000
##  1st Qu.:0.009204   1st Qu.:0.05668     1st Qu.:0.04420
##  Median :0.014494   Median :0.07194     Median :0.04895
##  Mean   :0.015072   Mean   :0.07853     Mean   :0.05080
##  3rd Qu.:0.019717   3rd Qu.:0.09476     3rd Qu.:0.05609
```

```
##  Max.   :0.061019   Max.   :0.27761     Max.   :0.11999
##  NA's   :3          NA's   :3           NA's   :3
##  edu_field_pers_protect_transp commute_transp_carpass commute_start_6am
##  Min.   :0.00000               Min.   :0.00000        Min.   :0.00000
##  1st Qu.:0.01485               1st Qu.:0.01951        1st Qu.:0.04423
##  Median :0.01969               Median :0.02297        Median :0.05714
##  Mean   :0.01957               Mean   :0.02351        Mean   :0.05662
##  3rd Qu.:0.02486               3rd Qu.:0.02826        3rd Qu.:0.06985
##  Max.   :0.03552               Max.   :0.04357        Max.   :0.10094
##  NA's   :3                     NA's   :3              NA's   :3
##  commute_start_9am     condo          worktype_selfemp  indigenous_ppl
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.02419   Min.   :0.000000
##  1st Qu.:0.03522   1st Qu.:0.03204   1st Qu.:0.05553   1st Qu.:0.009183
##  Median :0.04171   Median :0.08336   Median :0.06926   Median :0.019417
##  Mean   :0.04297   Mean   :0.12405   Mean   :0.07513   Mean   :0.036489
##  3rd Qu.:0.04900   3rd Qu.:0.16534   3rd Qu.:0.08961   3rd Qu.:0.035612
##  Max.   :0.09684   Max.   :0.85028   Max.   :0.22184   Max.   :0.700619
##  NA's   :3         NA's   :3         NA's   :3         NA's   :3
##   married_ppl     work_loc_foreign    can_citizen_ppl  non_citizen_ppl
##  Min.   :0.3193   Min.   :0.0000000   Min.   :0.6266   Min.   :0.00000
##  1st Qu.:0.4461   1st Qu.:0.0007752   1st Qu.:0.8536   1st Qu.:0.02517
##  Median :0.4818   Median :0.0013717   Median :0.9166   Median :0.06745
##  Mean   :0.4782   Mean   :0.0021423   Mean   :0.8980   Mean   :0.08630
##  3rd Qu.:0.5142   3rd Qu.:0.0022765   3rd Qu.:0.9549   3rd Qu.:0.12885
##  Max.   :0.6555   Max.   :0.0352995   Max.   :1.0754   Max.   :0.35330
##  NA's   :3        NA's   :3           NA's   :3        NA's   :3
##     TotalEV
##  Min.   :     4.32
##  1st Qu.:    40.81
##  Median :    65.11
##  Mean   :   432.68
##  3rd Qu.:   101.33
##  Max.   :102000.00
##
```

```
# Reading the dataset and removing the index . the summary describes the varibles and their distributio
```

```
# Find rows with any NA values
rows_with_na <- apply(data_ev, 1, function(x) any(is.na(x)))

# Display rows that have at least one NA value
data_ev_with_na <- data_ev[rows_with_na, ]

#data_ev_with_na$TotalEV <- ceiling(data_ev_with_na$TotalEV)

# Print the rows with NA values
print(data_ev_with_na)
```

```
## # A tibble: 4 x 29
##   FSA   med_fulltime_income income_100k_up med_total_household_income income_40k
##   <chr>               <dbl>          <dbl>                       <dbl>      <dbl>
## 1 K1A                    NA         0.0758                       73500      0.109
## 2 L4V                    NA        NA                               NA     NA
```

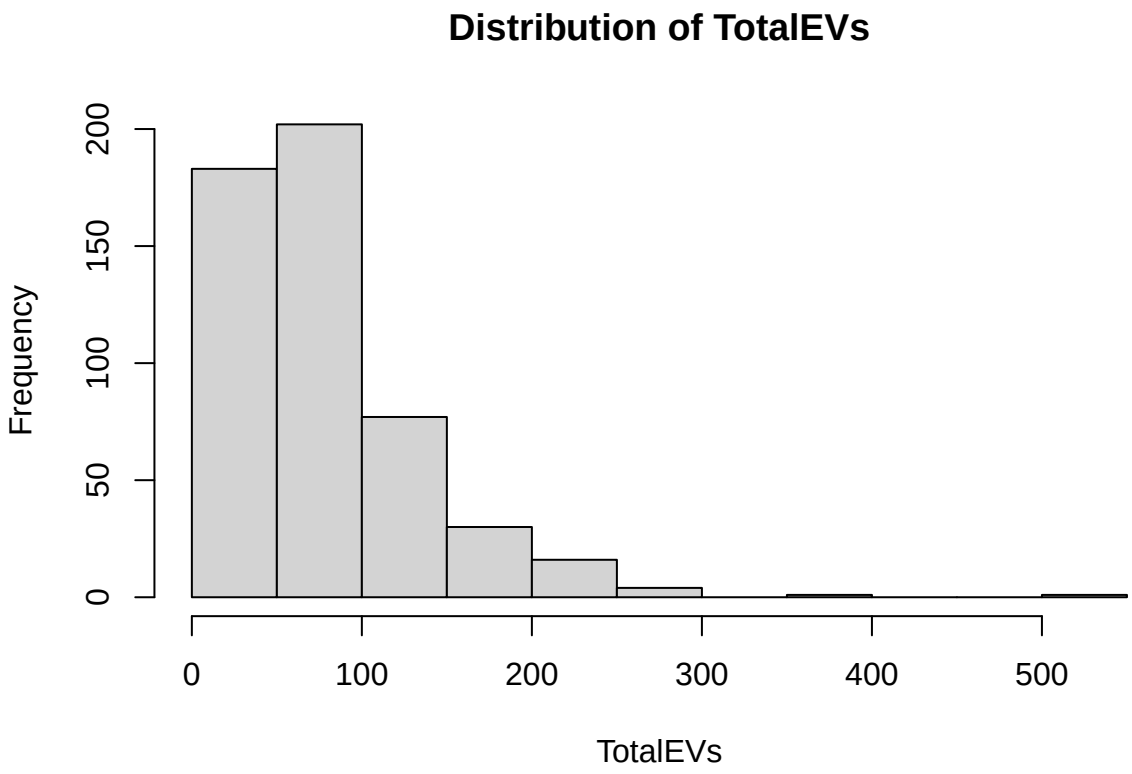
```
## 3 L5S                    NA        NA                                NA     NA
## 4 L5T                    NA        NA                                NA     NA
## # i 24 more variables: occup_cat_snr_mgmt <dbl>, occup_cat_busfin <dbl>,
## #   occup_ind_realestate <dbl>, work_loc_home <dbl>, work_loc_workplace <dbl>,
## #   work_loc_notfixed <dbl>, school_uni_degree <dbl>, school_hs <dbl>,
## #   commute_start_noon <dbl>, edu_field_science <dbl>,
## #   edu_field_bus_admin <dbl>, edu_field_health <dbl>,
## #   edu_field_pers_protect_transp <dbl>, commute_transp_carpass <dbl>,
## #   commute_start_6am <dbl>, commute_start_9am <dbl>, condo <dbl>, ...
```

```
data_ev_clean <- data_ev[complete.cases(data_ev[, -which(names(data_ev) == "TotalEV")]), ]

colSums(is.na(data_ev_clean))
```

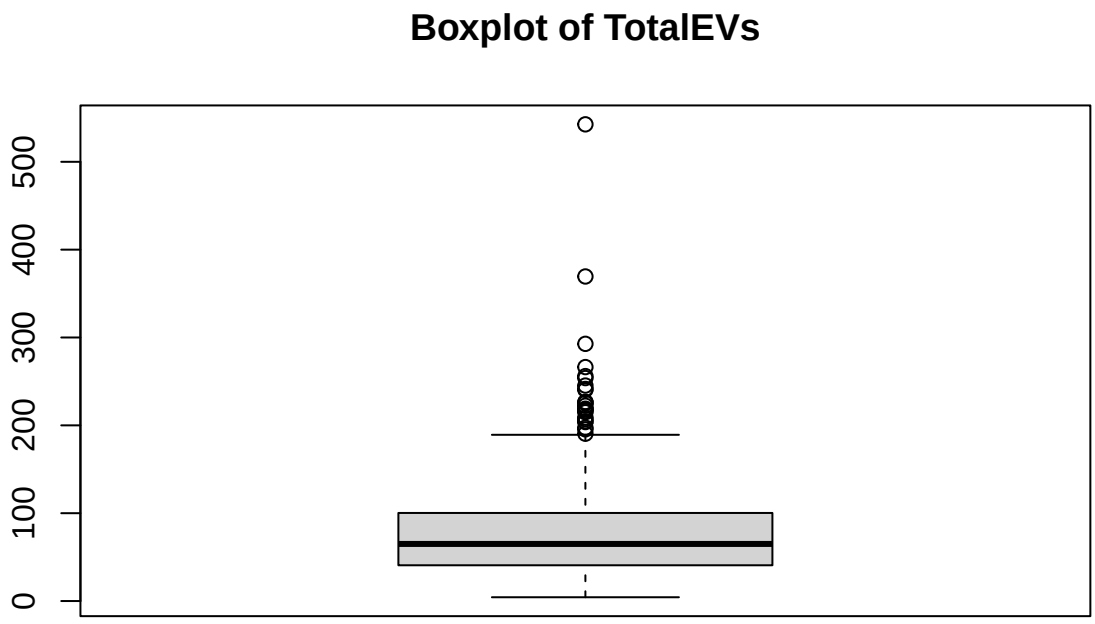
```
##                          FSA           med_fulltime_income
##                            0                             0
##                income_100k_up    med_total_household_income
##                            0                             0
##                    income_40k            occup_cat_snr_mgmt
##                            0                             0
##              occup_cat_busfin          occup_ind_realestate
##                            0                             0
##                 work_loc_home            work_loc_workplace
##                            0                             0
##             work_loc_notfixed             school_uni_degree
##                            0                             0
##                     school_hs            commute_start_noon
##                            0                             0
##             edu_field_science           edu_field_bus_admin
##                            0                             0
##              edu_field_health edu_field_pers_protect_transp
##                            0                             0
##        commute_transp_carpass             commute_start_6am
##                            0                             0
##             commute_start_9am                         condo
##                            0                             0
##              worktype_selfemp               indigenous_ppl
##                            0                             0
##                  married_ppl              work_loc_foreign
##                            0                             0
##               can_citizen_ppl               non_citizen_ppl
##                            0                             0
##                      TotalEV
##                            0
```

Now, we have completely no null values.

```
data_ev_clean$TotalEV <- as.numeric(as.character(data_ev_clean$TotalEV))

#data_ev_clean$TotalEV <- ceiling(data_ev_clean$TotalEV)

# EDA: Distribution of TOTALEV
hist(data_ev_clean$TotalEV, main = "Distribution of TotalEVs", xlab = "TotalEVs")
```




```r
# Boxplot for TOTALEV to check for outliers
boxplot(data_ev_clean$TotalEV, main = "Boxplot of TotalEVs")
```




```r
# Converting 'TOTALEV' into categorical variable 'EVCategory'
breaks <- quantile(data_ev_clean$TotalEV, probs=c(0, 0.33, 0.67, 1), na.rm = TRUE)
labels <- c("Low", "Medium", "High")
data_ev_clean$EVCategory <- cut(data_ev_clean$TotalEV, breaks=breaks, labels=labels, include.lowest = TR


# removing FSA and total ev variable
data_ev_clean <- data_ev_clean[, -which(names(data_ev_clean) %in% c("TotalEV", "FSA"))]


# Splitting the dataset into training and testing sets
set.seed(123)
indexes <- createDataPartition(data_ev_clean$EVCategory, p=0.7, list=FALSE)
trainData <- data_ev_clean[indexes, ]
testData <- data_ev_clean[-indexes, ]
```

```r
#install.packages("caret")

library(caret)
library(randomForest)

set.seed(123) # For reproducibility

ctrl <- caret::trainControl(method = "cv", number = 10)

rf_model <- caret::train(EVCategory ~ .,
```

```
                         data = trainData,
                         method = "rf",
                         trControl = ctrl,
                         tuneLength = 5)

print(rf_model)
```

```
## Random Forest
##
## 360 samples
##  27 predictor
##   3 classes: 'Low', 'Medium', 'High'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 324, 324, 324, 324, 324, 324, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.7389382  0.6083547
##    8    0.7224217  0.5835028
##   14    0.7363106  0.6045074
##   20    0.7474217  0.6211741
##   27    0.7502746  0.6253528
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

```
predictions <- predict(rf_model, newdata = testData)
confusionMatrix(predictions, testData$EVCategory)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Low Medium High
##     Low     43      8    0
##     Medium   7     40   12
##     High     1      4   39
##
## Overall Statistics
##
##                Accuracy : 0.7922
##                  95% CI : (0.7195, 0.8533)
##     No Information Rate : 0.3377
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6882
##
##  Mcnemar's Test P-Value : 0.167
##
## Statistics by Class:
##
```

```
##                      Class: Low Class: Medium Class: High
## Sensitivity              0.8431        0.7692      0.7647
## Specificity              0.9223        0.8137      0.9515
## Pos Pred Value           0.8431        0.6780      0.8864
## Neg Pred Value           0.9223        0.8737      0.8909
## Prevalence               0.3312        0.3377      0.3312
## Detection Rate           0.2792        0.2597      0.2532
## Detection Prevalence     0.3312        0.3831      0.2857
## Balanced Accuracy        0.8827        0.7915      0.8581
```

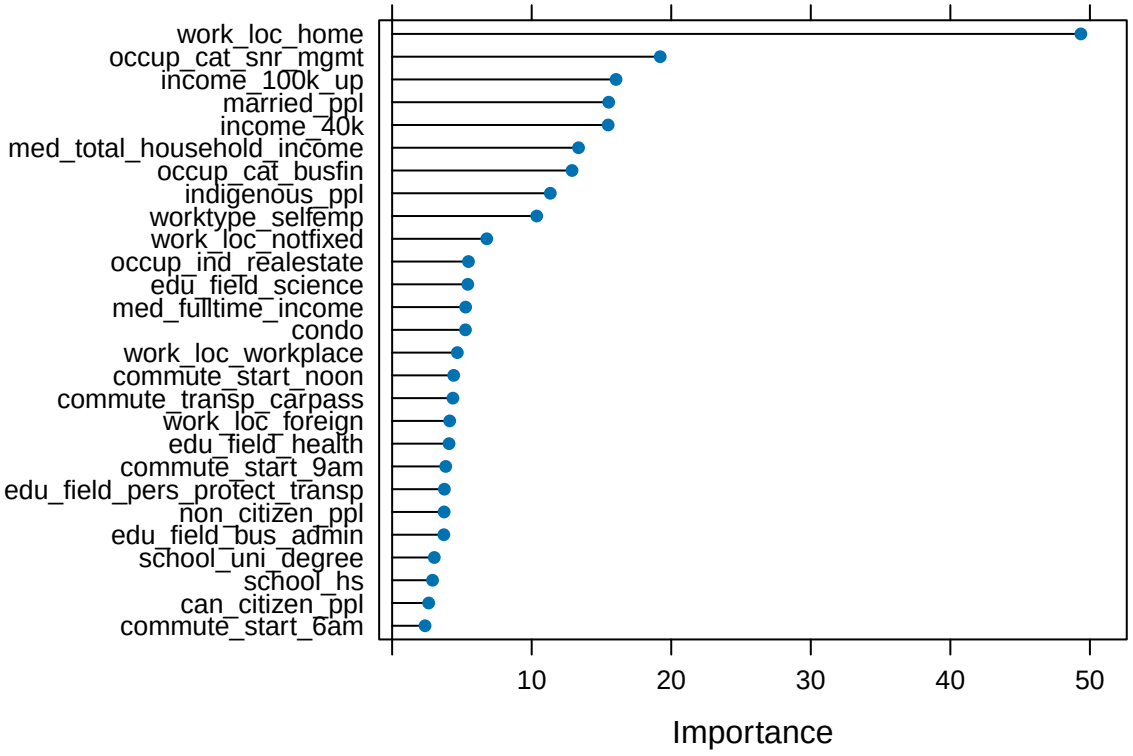
```r
# Making predictions on the test set
predictions <- predict(rf_model, testData)

# Calculate the accuracy
accuracy <- confusionMatrix(predictions, testData$EVCategory)
print(accuracy$overall["Accuracy"])
```

```
##  Accuracy
## 0.7922078
```

```r
importance <- varImp(rf_model, scale = FALSE)

# Print the variable importance
print(importance)
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 27)
##
##                             Overall
## work_loc_home                49.352
## occup_cat_snr_mgmt           19.208
## income_100k_up               16.043
## married_ppl                  15.521
## income_40k                   15.478
## med_total_household_income   13.357
## occup_cat_busfin             12.889
## indigenous_ppl               11.335
## worktype_selfemp             10.362
## work_loc_notfixed             6.783
## occup_ind_realestate          5.469
## edu_field_science             5.420
## med_fulltime_income           5.267
## condo                         5.255
## work_loc_workplace            4.672
## commute_start_noon            4.411
## commute_transp_carpass        4.356
## work_loc_foreign              4.124
## edu_field_health              4.079
## commute_start_9am             3.841
```

```r
# Plotting variable importance
plot(importance)
```




```r
#install.packages("xgboost")

# Install packages if you haven't already
# install.packages("xgboost")
# install.packages("caret")

library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 4.3.3
```

```
##
## Attaching package: 'xgboost'
```

```
## The following object is masked from 'package:plotly':
##
##     slice
```

```
## The following object is masked from 'package:dplyr':
##
##     slice
```

```r
library(caret)
library(Matrix) # For sparse matrix conversion, which is efficient for xgboost
```

```
## Warning: package 'Matrix' was built under R version 4.3.3
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
# Assuming you have a trainData and testData split

# Prepare the data for XGBoost
train_matrix <- sparse.model.matrix(EVCategory ~ . - 1, data = trainData)
dtrain <- xgb.DMatrix(data = train_matrix, label = as.numeric(trainData$EVCategory)-1) # Assuming EVCat

test_matrix <- sparse.model.matrix(EVCategory ~ . - 1, data = testData)
dtest <- xgb.DMatrix(data = test_matrix)

# Parameters for XGBoost
params <- list(
  booster = "gbtree",
  objective = "multi:softprob",
  num_class = length(unique(trainData$EVCategory)), # Number of classes
  eval_metric = "mlogloss", # Multiclass logloss
  eta = 0.1,
  max_depth = 6
)

# Train the model
nrounds <- 100 # Number of boosting rounds
xgb_model <- xgb.train(params = params, data = dtrain, nrounds = nrounds)

# Predictions
pred_probs <- predict(xgb_model, dtest)
pred_classes <- max.col(matrix(pred_probs, ncol = length(unique(trainData$EVCategory)), byrow = TRUE))-

# Convert numeric predictions back to factor levels
pred_classes_factor <- factor(pred_classes, levels = 0:(length(unique(trainData$EVCategory))-1), labels

# Evaluating the model
conf_mat <- confusionMatrix(pred_classes_factor, testData$EVCategory)
print(conf_mat)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Low Medium High
##     Low     43     11    0
##     Medium   6     35   10
```

```
##     High     2      6   41
##
## Overall Statistics
##
##                Accuracy : 0.7727
##                  95% CI : (0.6984, 0.8363)
##     No Information Rate : 0.3377
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6591
##
##  Mcnemar's Test P-Value : 0.2149
##
## Statistics by Class:
##
##                      Class: Low Class: Medium Class: High
## Sensitivity              0.8431        0.6731      0.8039
## Specificity              0.8932        0.8431      0.9223
## Pos Pred Value           0.7963        0.6863      0.8367
## Neg Pred Value           0.9200        0.8350      0.9048
## Prevalence               0.3312        0.3377      0.3312
## Detection Rate           0.2792        0.2273      0.2662
## Detection Prevalence     0.3506        0.3312      0.3182
## Balanced Accuracy        0.8682        0.7581      0.8631
```

```
accuracy <- conf_mat$overall['Accuracy']
print(accuracy)
```

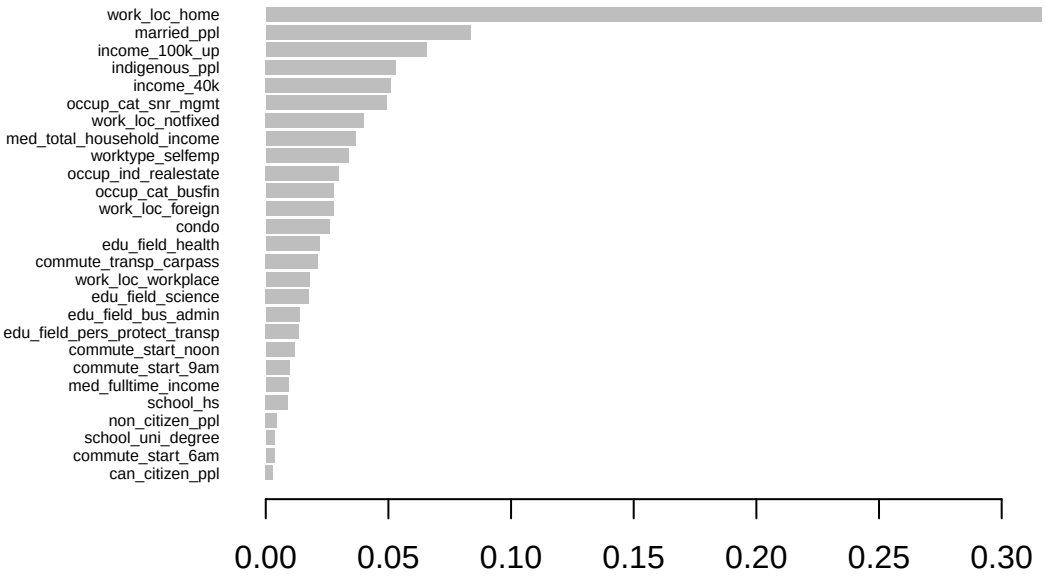
```
##  Accuracy
## 0.7727273
```

```
# Feature importance for XGBoost model
importance_matrix <- xgb.importance(feature_names = colnames(train_matrix), model = xgb_model)

# Print the feature importance
print(importance_matrix)
```

```
##                          Feature        Gain       Cover  Frequency
##                           <char>       <num>       <num>      <num>
##  1:                work_loc_home 0.316176662 0.138426717 0.05585928
##  2:                  married_ppl 0.083378207 0.069001913 0.06108866
##  3:               income_100k_up 0.065430681 0.060903272 0.03161398
##  4:               indigenous_ppl 0.053094893 0.083032668 0.05205610
##  5:                   income_40k 0.050985134 0.049079670 0.03541716
##  6:           occup_cat_snr_mgmt 0.049149965 0.057219712 0.05894937
##  7:            work_loc_notfixed 0.040115433 0.056685840 0.06203946
##  8:    med_total_household_income 0.036615778 0.045955349 0.04326123
##  9:             worktype_selfemp 0.033676204 0.051863248 0.05918707
## 10:         occup_ind_realestate 0.029890944 0.046001143 0.04444973
## 11:             occup_cat_busfin 0.027694384 0.060996092 0.05039220
## 12:             work_loc_foreign 0.027607269 0.028189335 0.04872831
## 13:                        condo 0.025889703 0.036398762 0.05062990
## 14:             edu_field_health 0.021828601 0.025055154 0.04492512
```

```
## 15:        commute_transp_carpass 0.021231639 0.014497497 0.02472070
## 16:            work_loc_workplace 0.017877433 0.018612051 0.04017114
## 17:             edu_field_science 0.017576572 0.017710261 0.03161398
## 18:           edu_field_bus_admin 0.013855936 0.014940453 0.01687663
## 19: edu_field_pers_protect_transp 0.013637132 0.015050037 0.02804849
## 20:            commute_start_noon 0.011713410 0.029998789 0.02923699
## 21:             commute_start_9am 0.009748228 0.021583126 0.03090088
## 22:           med_fulltime_income 0.009211064 0.013089469 0.01996672
## 23:                    school_hs 0.009119265 0.014888943 0.02115522
## 24:               non_citizen_ppl 0.004571507 0.012336504 0.02044212
## 25:             school_uni_degree 0.003530656 0.006392598 0.01164725
## 26:             commute_start_6am 0.003485685 0.005211243 0.01283575
## 27:               can_citizen_ppl 0.002907617 0.006880156 0.01378655
##                         Feature        Gain       Cover  Frequency
```

```r
# Plot the feature importance
xgb.plot.importance(importance_matrix)
```




```r
# Load necessary libraries
library(mlr)
```

```
## Warning: package 'mlr' was built under R version 4.3.3
```

```
## Loading required package: ParamHelpers
```

```
## Warning: package 'ParamHelpers' was built under R version 4.3.3
```

```
## Warning message: 'mlr' is in 'maintenance-only' mode since July 2019.
## Future development will only happen in 'mlr3'
## (<https://mlr3.mlr-org.com>). Due to the focus on 'mlr3' there might be
## uncaught bugs meanwhile in {mlr} - please consider switching.
```

```
##
## Attaching package: 'mlr'
```

```
## The following object is masked from 'package:e1071':
##
##     impute
```

```
## The following object is masked from 'package:caret':
##
##     train
```

```
library(randomForest)
library(xgboost)

# Define the classification task
task <- makeClassifTask(data = trainData, target = "EVCategory")
```

```
## Warning in makeTask(type = type, data = data, weights = weights, blocking =
## blocking, : Provided data is not a pure data.frame but from class tbl_df, hence
## it will be converted.
```

```
# Define base learners
learners <- list(
  makeLearner("classif.randomForest", id = "rf"),
  makeLearner("classif.xgboost", id = "xgb")
)

# Use a Random Forest classifier as the super learner
# This is known to support multiclass classification
superLearner <- makeLearner("classif.randomForest", id = "superLearner")

# Create the stacked learner
stackedLearner <- makeStackedLearner(base.learners = learners, super.learner = superLearner, method = "

# Train the stacked model
model_boost <- train(stackedLearner, task)

pred <- predict(model_boost, newdata = testData) # Corrected, testData should be a data.frame
```

```
## Warning in predict.WrappedModel(model_boost, newdata = testData): Provided data
## for prediction is not a pure data.frame but from class tbl_df, hence it will be
## converted.
```

```r
# If you have the true labels available for testData, you can evaluate the model's performance
true_labels <- testData$EVCategory # Assuming the true labels are stored in testData

# Convert predictions to a factor or appropriate format if needed
predicted_labels <- as.factor(pred$data$response)

# Calculate accuracy or other performance metrics
library(caret)
conf_mat_boost <- confusionMatrix(predicted_labels, true_labels)
print(conf_mat_boost)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Low Medium High
##     Low     43      8    0
##     Medium   7     40   10
##     High     1      4   41
##
## Overall Statistics
##
##                Accuracy : 0.8052
##                  95% CI : (0.7337, 0.8645)
##     No Information Rate : 0.3377
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7077
##
##  Mcnemar's Test P-Value : 0.3033
##
## Statistics by Class:
##
##                      Class: Low Class: Medium Class: High
## Sensitivity              0.8431        0.7692      0.8039
## Specificity              0.9223        0.8333      0.9515
## Pos Pred Value           0.8431        0.7018      0.8913
## Neg Pred Value           0.9223        0.8763      0.9074
## Prevalence               0.3312        0.3377      0.3312
## Detection Rate           0.2792        0.2597      0.2662
## Detection Prevalence     0.3312        0.3701      0.2987
## Balanced Accuracy        0.8827        0.8013      0.8777
```

```r
accuracy <- conf_mat_boost$overall['Accuracy']
print(accuracy)
```

```
##  Accuracy
## 0.8051948
```

G Results

Exploratory Data Analysis:

Our analysis began by closely examining the dataset to discern patterns and relationships, employing scatterplots to illuminate the interactions between the independent variables and the response variable `TotalEV`. These visual examinations hinted at a positive correlation between income-related variables and `TotalEV`, suggesting that areas with higher incomes tended to also exhibit increased `TotalEV` values. The plots revealed a densely packed cluster of data points at lower income levels, with a noticeable spread at higher income tiers, indicating income's significant but not exclusive influence on `TotalEV`.

Subsequent scrutiny of occupational variables, like senior management and business and finance categories, revealed a potential correlation with `TotalEV`, albeit with overlapping categories that suggest additional contributing factors. The predictive capacity of demographic and real estate variables, such as marital status and the proportion of people working from home, surfaced in the plots as nuanced indicators that warrant deeper investigation to unravel the complex socio-economic web influencing `TotalEV`.

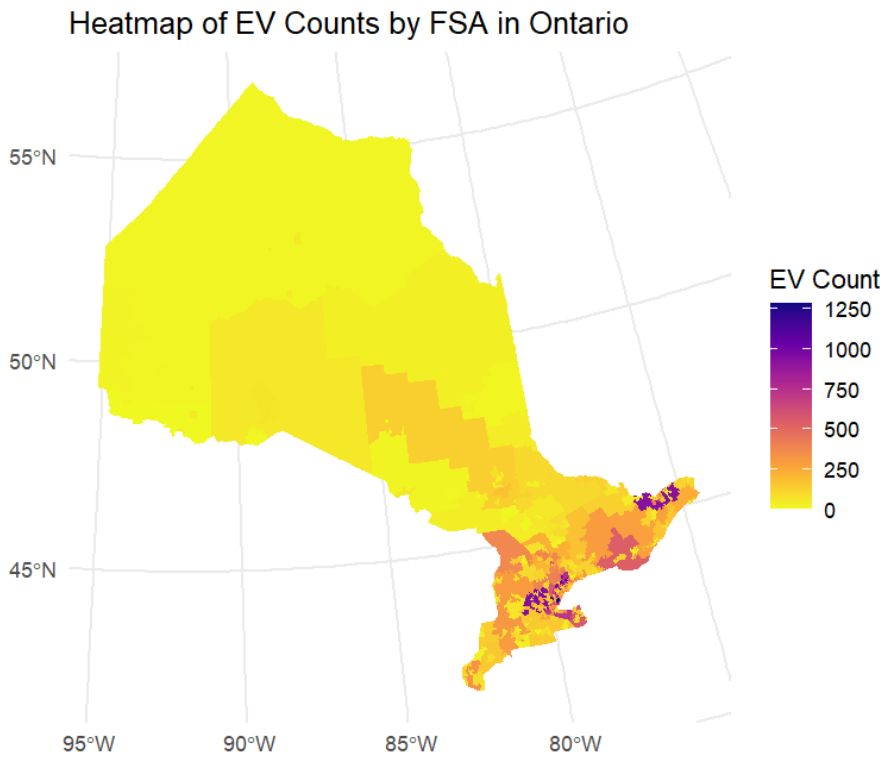


Figure 2: Data Flow pipeline

In the first map, the color gradient from yellow to dark purple indicates low to high densities of EV counts, respectively. Areas with the highest concentrations of electric vehicles are depicted in darker shades. There is a significant geographic variation in EV distribution across Ontario. The southern part of the province, shows the highest concentration of electric vehicles. This area includes major urban centers, which from the map seems to have higher adoption rates for EVs due to better infrastructure and greater environmental awareness. The vast majority of northern Ontario shows a very low density of EVs which could be due to a variety of factors such as less developed charging infrastructure, lower population density, or different transportation needs and preferences.

The second map overlays black dots to indicate the locations of individual EV stations. There appears to be a strong correlation between the presence of EV charging stations and higher counts of electric vehicles, particularly in southern Ontario. The dense clusters of black dots coincide with the darker areas on the heatmap suggesting the the availability of charging infrastructure is a likely factor in EV adoption. The

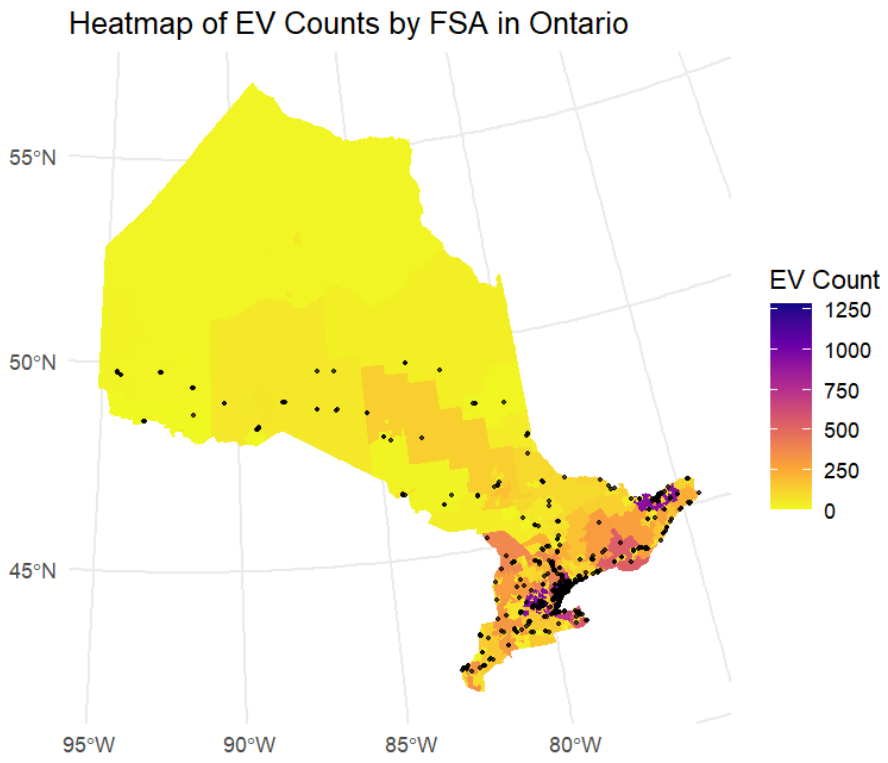


Figure 3: Data Flow pipeline

visualization supports the hypothesis that the presence of charging stations influences the adoption of electric vehicles. For policymakers and businesses, this data could justify investments in charging infrastructure in areas that are currently underserved to promote EV adoption.

Data Normalization:

There are 2,631 characteristics (rows) for each FSA, resulting in 1,370,751 total rows of interest. We selected 143 characteristics, corresponding to the domains we identified as possibly yielding features from which EV ownership/registration may be predicted, as well as those necessary to normalize other values. After selecting these, the table was transposed to a format with one row per FSA, and each column representing a selected feature.

The EV and Census Profile datasets were joined on the FSA identifiers, resulting in 517 observations (FSAs present in both data sets) with 147 columns. Four FSAs with very small or no populations were removed. Statistics referring to counts of individual people were divided by the overall population of the area, while those referring to counts of dwellings (homes) were divided by the number of dwellings in the area. The number of registered EVs were divided by the population and multiplied by 10,000 to yield the number of registered EVs per 10,000 people.

After assembling our data and normalizing it over the population, we ran a simple correlation check between the possible features and the (normalized) TotalEV value. We selected 26 features from across multiple domains with the highest correlations (positive or negative) in their categories. Note that only "med_fulltime_income" would qualify as being particularly high, but other features may still contribute in some ways.

The source data is relatively high-quality, and requires little in the way of cleaning, beyond the normalization we have already done. Missing values only appear to exist for those regions with very small populations; we

have removed four FSAs from consideration as a result.

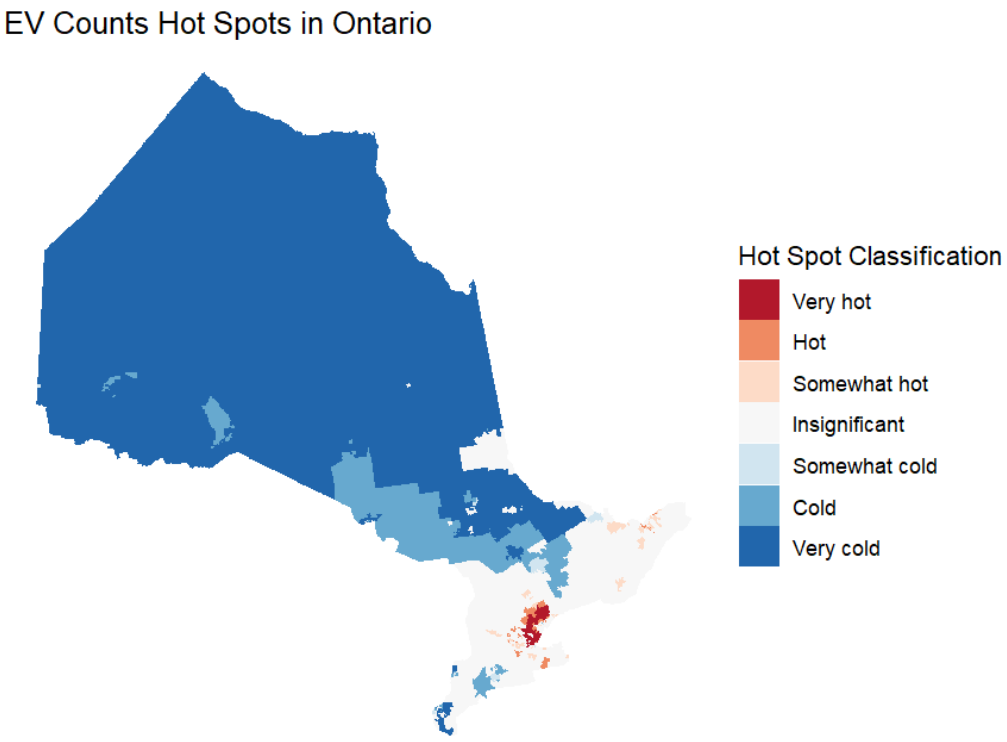
Research Question 1:

In our exploratory analysis of the dataset, we crafted scatter plots which allowed us to visually assess the relationships between the independent variables and the response variable, guiding us toward the appropriate modeling approach. Delving into the distribution of the response variable, histograms provided us with initial insights into its characteristics. Through this process, we determined that a Generalized Linear Model (GLM) was a fitting choice, and after comparing different distribution families, the Negative Binomial distribution emerged as the most suitable, as evidenced by its minimal AIC value compared to the Poisson and Quasi-Poisson distributions. The choice of the Negative Binomial model was further substantiated by its ability to account for overdispersion—a common issue with count data—more effectively than the Poisson model. Additionally, the plots of predicted versus actual values demonstrated the model's capacity to closely estimate the actual data points, affirming its predictive validity, particularly for data points with higher values. Our analysis underscores the value of a meticulous model selection process, with the Negative Binomial GLM proving to be a robust method for handling the nuances of our data.

Research Question 2:

GLM
A Negative Binomial GLM was fit with the count of EVs as the response and the count of the EV stations as the explanatory variable. The residual diagnostic plot showed no major issues as well as the QQ plot where majority of the data fell on the line. For coefficients, the intercept had a value of 5.16845 with a very small standard error and it is statistically significant (indicated by a p-value less than 2e-16). This coefficient represents the log count of electric vehicles when the count of EV stations is zero. This roughly translates to about 176 EVs when the count of EV stations is zero. This may be to people owning their private means of charging their electric vehicles and not relying on public charging infrastructure. The coefficient for EV station count is 0.11483, suggesting that for every one-unit increase in the count of Ev stations, the log count of electric vehicles is expected to increase by this amount. This coefficient is also statistically significant (p-value = 1.99e-05) which suggests there is a strong statistical evidence that the number of EV stations is positively associated with the number of electric vehicles.

Hot Spot Analysis



To further validate the EDA we performed hotspot analysis on the counts of EVs across FSAs in Ontario. The hotspot map illustrates the distribution of electric vehicle counts by Forward Sortation Area across Ontario, using color coding to represent areas of varying EV concentration. High concentration areas (very hot spots), depicted in dark red, indicate FSAs with the highest EV counts. These areas seem to be urban centers with greater access to EV infrastructure for EV owners. These regions may also have higher income levels, allowing for greater adoption of EV technology. The areas marked as "Hot" and "Somewhat hot" in lighter red and orange shades, respectively, suggest moderate counts of EVs. These could be suburban areas or smaller towns where there is some infrastructure and interest in EVs, but where certain barriers to full adoption might still exist. White areas denote FSAs where EV counts don't significantly deviate from the average. These areas could have a balanced mix of conditions both favoring and deterring EV adoption. The areas in shades of blue, which show up as "Cold" and "Very cold," suggest low EV counts. This could be due to a lack of infrastructure, lower environmental initiatives, economic barriers, or other factors that might discourage EV ownership. As far as policy implications go, the government could look into providing incentives or support programs, development of charging infrastructure, particularly in "Cold" and "Somewhat cold" areas to boost adoption rates.

Research Question 3

In this phase of our study, we applied Random Forest, a powerful tree-based ensemble learning method, to predict the categorization of electric vehicles (EVs) based on demographic variables. The Random Forest model is particularly well-suited for this task due to its capability to handle high-dimensional data and its robustness against overfitting, which is common in decision tree models. By aggregating the predictions of numerous decision trees, Random Forest improves prediction accuracy and model interpretability.

Random Forest Model Training and Validation

We utilized the caret package in R for model training and evaluation, setting a seed for reproducibility and specifying a 10-fold cross-validation in the training control settings to assess model performance reliably. The model was trained on the dataset, excluding the FSA and TotalEV variables to prevent data leakage and to focus on demographic factors.

Upon training, the model's performance was evaluated on a test dataset, resulting in an accuracy of approximately **0.7987**. This indicates that the model can correctly predict the EV category (Low, Medium, High) based on demographic variables with a fairly high level of reliability.

Variable Importance One of the key strengths of Random Forest is its ability to quantify the importance of each variable in making predictions. This feature is invaluable for understanding which demographic factors most significantly influence EV adoption. The varImp function was used to derive the importance scores for all demographic variables in the dataset, scaled by the increase in prediction accuracy they provide.

The results revealed that the variable **work_loc_home** (representing the proportion of individuals working from home) is the most influential predictor of EV categories, followed by occup_cat_snr_mgmt (the proportion of individuals in senior management occupations), and income_100k_up (the proportion of individuals with income over 100k). These findings suggest that areas with a higher proportion of remote workers, senior management professionals, and high-income earners are more likely to have a higher category of EV adoption.

This analysis not only sheds light on the demographic characteristics associated with higher EV adoption rates but also provides valuable insights for policymakers and stakeholders interested in promoting EV usage. By understanding the demographic variables that most strongly predict EV adoption, targeted strategies and incentives can be developed to encourage EV uptake in specific communities or demographic groups.

The 5 utmost important variables according to random forest

work_loc_home
occup_cat_snr_mgmt
income_100k_up
married_ppl
income_40k

XGBoost Model Training and Evaluation To leverage the capabilities of XGBoost within the R environment, we prepared our dataset by converting it into a sparse matrix format, optimizing for efficiency given the algorithm's handling of sparse data. Our model was trained using parameters fine-tuned to balance model complexity and learning rate, aiming for a generalized model with high predictive accuracy.

After training, the model's performance was evaluated on the test dataset, achieving an accuracy of approximately **0.7727**. This result highlights the model's effectiveness in classifying EV categories based on demographic variables, providing a solid foundation for predictive insights.

Variable Importance Analysis An essential aspect of our analysis involved understanding which demographic variables most significantly influence the classification of EV categories. XGBoost offers a built-in method for assessing feature importance, which we utilized to identify the top predictors in our model.

The analysis revealed that **work_loc_home**, indicating the proportion of people working from home, emerged as the most significant predictor. This variable was followed by married_ppl, income_100k_up, and indigenous_ppl, among others. These findings suggest that lifestyle and socio-economic factors play crucial roles in determining EV adoption levels.

Ensemble Learning we leveraged the power of ensemble methods by combining Random Forest and XGBoost models through a boosting method. This approach, known for enhancing model performance by aggregating predictions from multiple models, allows for a more robust and accurate predictive framework, particularly in complex classification tasks like ours.

Ensemble Learning with Boosting

We initiated our ensemble strategy by defining base learners - a Random Forest classifier and an XGBoost model - each known for their predictive capabilities in handling structured data. The Random Forest model contributes by capturing complex, nonlinear relationships through multiple decision trees, while the XGBoost model adds gradient boosting capabilities, focusing on optimizing model performance and addressing model bias.

To harmonize these models' strengths, we employed a super learner approach, choosing a Random Forest classifier as the super learner. This choice was strategic, given Random Forest's efficacy in managing multiclass classification problems and its ability to reduce overfitting through its ensemble nature.

Training and Predictive Accuracy
Upon training the stacked ensemble model on our classification task, which aimed to predict EV adoption categories based on demographic variables, we evaluated its performance on a separate test dataset. The predictions were converted to the appropriate format, and accuracy was calculated, yielding a notable accuracy score of approximately **0.8052**. This result underscores the ensemble model's effectiveness in accurately classifying EV categories, illustrating the added value of combining multiple learning algorithms for enhanced predictive power.

Final Implications and Insights From Tree Based Models

The superior accuracy achieved by the ensemble model underscores the potential of leveraging combined model strengths to tackle complex predictive challenges. This approach not only amplifies the individual models' capabilities but also offers a deeper understanding of the underlying patterns within the data, providing a more nuanced view of the factors driving EV adoption across different demographics.

For policymakers and stakeholders in the EV industry, these insights are invaluable. They highlight the multifaceted nature of EV adoption and the importance of tailored strategies that address specific demographic characteristics. The enhanced predictive accuracy of our ensemble model offers a more reliable basis for decision-making, supporting targeted interventions to promote sustainable transportation.

H Conclusion

The burgeoning electric vehicle (EV) market in Canada, as reflected in the notable increase in EV registrations, marks a paradigm shift towards environmentally conscious transportation. This shift is not only in alignment with Canada's ambitious climate goals but also mirrors the growing public appetite for sustainable vehicle options. Our exploratory data analysis has unearthed a clear positive correlation between the presence of EV charging infrastructure and the uptick in EV adoption, particularly in densely populated urban areas.

The meticulous application of a Negative Binomial Generalized Linear Model has yielded statistically significant insights, underscoring the importance of EV charging stations in bolstering EV prevalence. The positive association between charging infrastructure and EV counts emphasizes the necessity for strategic policy interventions. Furthermore, the hotspot analysis vividly illustrates the spatial disparity in EV adoption across Ontario, pinpointing potential areas for infrastructural enhancement.

Policy recommendations emanating from this research should center on bolstering infrastructural development, particularly in areas identified as cold spots, to democratize access to EVs. Incentive programs and educational initiatives may serve to alleviate the barriers to adoption, promoting a widespread shift to EVs. Moreover, future research should aim to dissect the layers of demographic, economic, and behavioral factors influencing this transition, tailoring solutions to the nuanced needs of diverse regions.

In conclusion, this study confirms that EV adoption in Ontario is not just a function of consumer choice but is intricately linked to the availability of supportive infrastructure. As Canada strides towards its 2035 phase-out goal for gas vehicles, the insights gleaned herein advocate for a concerted effort in infrastructure expansion, public education, and policy innovation. The pursuit of such endeavors is pivotal to catalyze the transition to electric mobility, fostering a sustainable future that resonates with national climate aspirations and the environmental ethos of the Canadian populace.

References

Data & Software

Emerging Technologies Office. (2024). Electric Vehicles in Ontario – By Forward Sortation Area—Q4 2023 [dataset]. Ontario Data Catalogue. https://data.ontario.ca/dataset/electric-vehicles-in-ontario-by-forward-sortation-area/resource/dca5bef6-df38-4c45-be73-62e71b243d3d

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12(85), 2825–2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

Statistics Canada. (2022, September 21). Census Forward Sortation Area Boundary File, Census year 2021 [shapefile]. https://www150.statcan.gc.ca/n1/en/catalogue/92-179-X

Statistics Canada. (2022, February 9). Census Profile Downloads, 2021 [dataset]. https://www12.statcan.gc.ca/census-recensement/2021/dp-pd/prof/details/download-telecharger.cfm?Lang=E

Referenced Works

Blair, N. (2024, January). Electric Vehicle Adoption Statistics in Canada. Made in CA. https://madeinca.ca/electric-vehicle-adoption-statistics-canada/

Chen, C. F., de Rubens, G. Z., Noel, L., Kester, J., & Sovacool, B. K. (2020). Assessing the socio-demographic, technical, economic and behavioral factors of Nordic electric vehicle adoption and the influence of vehicle-to-grid preferences. Renewable and Sustainable Energy Reviews, 121, 109692.https://www.sciencedirect.com/science/article/abs/pii/S1364032119308974

Cousins, B. (2023, December). What does 2024 have in store for the EV industry? Bloomberg BNN.https://www.bnnbloomberg.ca/what-does-2024-have-in-store-for-the-ev-industry-1.2014925

Egbue, O., Long, S., & Samaranayake, V. A. (2017). Mass deployment of sustainable transportation: evaluation of factors that influence electric vehicle adoption. Clean Technologies and Environmental Policy, 19, 1927-1939.https://link.springer.com/article/10.1007/s10098-017-1375-4

RPubs—R Tutorial: Hotspot Analysis Using Getis Ord Gi. (n.d.). Retrieved April 10, 2024, from https://rpubs.com/heatherleeleary/hotspot_getisOrd_tut