Code - A bit more organized

Dylan Koproski

2025-01-28

Libraries

```
library(tidyverse)
```

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

```
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
##
## The following object is masked from 'package:purrr':
##
##     flatten
```

```
library(readxl)
library(rsample)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(VGAM)
```

```
## Loading required package: stats4
## Loading required package: splines
##
## Attaching package: 'VGAM'
##
## The following object is masked from 'package:caret':
```

```
##
##     predictors
```

```
library(COMPoissonReg)
```

```
## Loading required package: Rcpp
##
## Attaching package: 'Rcpp'
##
## The following object is masked from 'package:rsample':
##
##     populate
##
## Loading required package: numDeriv
##
## Attaching package: 'COMPoissonReg'
##
## The following object is masked from 'package:VGAM':
##
##     get.offset
```

```
library(pscl)
```

```
## Classes and Methods for R originally developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University (2002-2015),
## by and under the direction of Simon Jackman.
## hurdle and zeroinfl functions by Achim Zeileis.
```

```
library(lme4)
```

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

```
library(zipcodeR)
library(maps)
```

```
##
## Attaching package: 'maps'
##
## The following object is masked from 'package:purrr':
##
##     map
```

```
library(MASS)
```

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

```
library(usmap)
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
conflicted::conflict_prefer("select", "dplyr")
```

```
## [conflicted] Will prefer dplyr::select over any other package.
```

```
conflicted::conflict_prefer("map", "purrr")
```

```
## [conflicted] Will prefer purrr::map over any other package.
```

```
conflicted::conflict_prefer("filter", "dplyr")
```

```
## [conflicted] Will prefer dplyr::filter over any other package.
```

Preprocessing

```
# Load data
df_visitor = read_csv("mobility.csv")
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 24583 Columns: 52
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (30): placekey, parent_placekey, safegraph_brand_ids, location_name, br...
## dbl  (14): naics_code, latitude, longitude, phone_number, wkt_area_sq_meters...
## lgl   (3): enclosed, is_synthetic, includes_parking_lot
## dttm  (2): date_range_start, date_range_end
## date  (3): opened_on, closed_on, tracking_closed_since
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df_census = read_csv("tract_census.csv", skip = 1) |>
  janitor::clean_names()
```

```
## New names:
## * `` -> `...459`
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
```

```
##   problems(dat)
```

```
## Rows: 85395 Columns: 459
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (272): Geography, Geographic Area Name, Estimate!!Total!!Total populatio...
## dbl (186): Estimate!!Total!!Total population, Margin of Error!!Total!!Total ...
## lgl   (1): ...459
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df_tract_zip = read_excel("tract_zip.xlsx")



# Temporary restriction to NYC
state_county_code_str = c(36005, 36047, 36061, 36081, 36085)

# Process visitor data - none missing
filtered_df_visitor = df_visitor |>
  mutate(first_five_digits = substr(poi_cbg, 1, 5)) |>
 # filter(first_five_digits %in% state_county_code_str) |>
  # Dropping missing values for home cbg
  filter(!is.na(visitor_home_aggregation)) |>
  mutate(identifier = row_number()) |>
  mutate(poi_zip = postal_code) |>
  mutate(visitor_home_aggregation = map(visitor_home_aggregation, ~fromJSON(as.character(.)))) |>
  select(location_name, date_range_start, date_range_end, visitor_home_aggregation, top_category, identi
  unnest_longer(visitor_home_aggregation) |>
  rename(visitor_census_tract = visitor_home_aggregation_id, visitors = visitor_home_aggregation) |>
  mutate(visitors = if_else(visitors == 4, 3, visitors)) |>
  mutate(poi_lat = latitude,
         poi_long = longitude)

# Census data processing
df_census = df_census |>
  rowwise() |>
  mutate(cbg = str_sub(geography, -11)) |>
#873 locations have an estimated 0 people, we should exclude these.
  filter(estimate_total_total_population > 0)

# Age group proportions in census data, separate into 3 age groups
filtered_df_census_totals =
  df_census |>
  rowwise() |>
  select(estimate_total_total_population, cbg, geographic_area_name, starts_with("estimate")) |>
  mutate(
    total_under_18 = sum(estimate_total_total_population_age_under_5_years,
                         estimate_total_total_population_age_5_to_9_years,
                         estimate_total_total_population_age_10_to_14_years,
                         estimate_total_total_population_age_15_to_19_years) / estimate_total_total_popu

    total_19_65 = sum(estimate_total_total_population_age_20_to_24_years,
                      estimate_total_total_population_age_25_to_29_years,
```

```
                      estimate_total_total_population_age_30_to_34_years,
                      estimate_total_total_population_age_35_to_39_years,
                      estimate_total_total_population_age_40_to_44_years,
                      estimate_total_total_population_age_45_to_49_years,
                      estimate_total_total_population_age_50_to_54_years,
                      estimate_total_total_population_age_55_to_59_years,
                      estimate_total_total_population_age_60_to_64_years,
                      estimate_total_total_population_age_65_to_69_years) / estimate_total_total_popula

    total_65_plus = sum(estimate_total_total_population_age_70_to_74_years,
                        estimate_total_total_population_age_75_to_79_years,
                        estimate_total_total_population_age_80_to_84_years,
                        estimate_total_total_population_age_85_years_and_over) / estimate_total_total_p
  ) |>
  rename("total" = estimate_total_total_population) |>
  select(cbg, geographic_area_name, total, total_under_18, total_19_65, total_65_plus)

# Define primary ZIP code per census tract
primary_tract_zip = df_tract_zip |>
  group_by(tract) |>
  summarize(zip = min(zip))  # Selects the minimum ZIP as primary for simplicity

# Merge filtered_df_census_totals with filtered_df_visitor
merged_df = filtered_df_visitor |>
  inner_join(filtered_df_census_totals, by = c("visitor_census_tract" = "cbg")) |>
  mutate(
    visitors_under_18 = visitors * total_under_18,
    visitors_19_65 = visitors * total_19_65,
    visitors_65_plus = visitors * total_65_plus
  )

# Map primary ZIP codes by merging with primary_tract_zip on the census tract
final_df = merged_df |>
  left_join(primary_tract_zip, by = c("visitor_census_tract" = "tract")) |>
  select(location_name, date_range_start, date_range_end, top_category,
         identifier, poi_cbg, poi_zip, visitors, visitors_under_18,
         visitors_19_65, visitors_65_plus, zip, poi_long, poi_lat) |>
  mutate(visitor_zip = zip)

vis_zip_lat_long = geocode_zip(final_df$visitor_zip)

final_df = final_df |>
  left_join(vis_zip_lat_long, by = join_by(visitor_zip == zipcode)) |>
  mutate(vis_lat = lat,
         vis_long = lng) |>
  select(-lat, -lng)

# Rounding, can be adjusted at will
final_df = final_df |>
  mutate(visitors_under_18 = ceiling(visitors_under_18),
         visitors_19_65 = ceiling(visitors_19_65),
         visitors_65_plus = ceiling(visitors_65_plus)) |>
  mutate(total_visitors = visitors_under_18 + visitors_19_65 + visitors_65_plus)
```

```r
# There are 2 rows in the above data that have missing values, the visitor zip is missing. We should fi

# Long dataframe for modelling purposes
df_long = final_df |>
  pivot_longer(
    cols = starts_with("visitors_"),
    names_to = "age_group",
    names_prefix = "visitors_",
    values_to = "visitor_count"
  ) |>
  mutate(top_category = factor(top_category),
         age_group = factor(age_group, levels = c("under_18", "19_65", "65_plus")))

# Data quality looks good, 2 missing zip codes may need to be handled, but data is large enough maybe d
```

Category definitions

```r
all_cat = c(
  "Accounting, Tax Preparation, Bookkeeping, and Payroll Services",
  "Activities Related to Credit Intermediation",
  "Activities Related to Real Estate",
  "Advertising, Public Relations, and Related Services",
  "Agencies, Brokerages, and Other Insurance Related Activities",
  "Architectural, Engineering, and Related Services",
  "Automotive Parts, Accessories, and Tire Stores",
  "Automotive Repair and Maintenance",
  "Bakeries and Tortilla Manufacturing",
  "Beer, Wine, and Liquor Stores",
  "Book Stores and News Dealers",
  "Building Equipment Contractors",
  "Building Finishing Contractors",
  "Building Material and Supplies Dealers",
  "Child Day Care Services",
  "Clothing Stores",
  "Consumer Goods Rental",
  "Couriers and Express Delivery Services",
  "Depository Credit Intermediation",
  "Drinking Places (Alcoholic Beverages)",
  "Drycleaning and Laundry Services",
  "Electronic and Precision Equipment Repair and Maintenance",
  "Electronics and Appliance Stores",
  "Elementary and Secondary Schools",
  "Florists",
  "Furniture Stores",
  "Gasoline Stations",
  "General Medical and Surgical Hospitals",
  "General Merchandise Stores, including Warehouse Clubs and Supercenters",
  "Glass and Glass Product Manufacturing",
  "Grocery Stores",
  "Health and Personal Care Stores",
  "Home Furnishings Stores",
  "Investigation and Security Services",
```

```
  "Jewelry, Luggage, and Leather Goods Stores",
  "Justice, Public Order, and Safety Activities",
  "Legal Services",
  "Machinery, Equipment, and Supplies Merchant Wholesalers",
  "Museums, Historical Sites, and Similar Institutions",
  "Offices of Dentists",
  "Offices of Other Health Practitioners",
  "Offices of Physicians",
  "Offices of Real Estate Agents and Brokers",
  "Other Amusement and Recreation Industries",
  "Other Financial Investment Activities",
  "Other Miscellaneous Manufacturing",
  "Other Miscellaneous Store Retailers",
  "Other Personal Services",
  "Other Professional, Scientific, and Technical Services",
  "Other Schools and Instruction",
  "Other Specialty Trade Contractors",
  "Personal and Household Goods Repair and Maintenance",
  "Personal Care Services",
  "Printing and Related Support Activities",
  "Promoters of Performing Arts, Sports, and Similar Events",
  "Radio and Television Broadcasting",
  "Religious Organizations",
  "Restaurants and Other Eating Places",
  "Shoe Stores",
  "Sound Recording Industries",
  "Special Food Services",
  "Specialized Design Services",
  "Specialty (except Psychiatric and Substance Abuse) Hospitals",
  "Specialty Food Stores",
  "Sporting Goods, Hobby, and Musical Instrument Stores",
  "Support Activities for Road Transportation",
  "Technical and Trade Schools",
  "Transit and Ground Passenger Transportation",
  "Traveler Accommodation",
  "Warehousing and Storage",
  "Wired and Wireless Telecommunications Carriers"
)

medical_services = c(
  "General Medical and Surgical Hospitals",
  "Health and Personal Care Stores",
  "Offices of Dentists",
  "Offices of Other Health Practitioners",
  "Specialty (except Psychiatric and Substance Abuse) Hospitals",
  "Offices of Physicians"
)

essential_services = c(
  "General Medical and Surgical Hospitals",
  "Health and Personal Care Stores",
  "Pharmacies and Drug Stores",
  "Grocery Stores",
```

```
  "Gasoline Stations",
  "Depository Credit Intermediation",
  "Public Transport Hubs",
  "Government Offices"
)

retail_shopping = c(
  "General Merchandise Stores, including Warehouse Clubs and Supercenters",
  "Clothing Stores",
  "Shoe Stores",
  "Jewelry, Luggage, and Leather Goods Stores",
  "Electronics and Appliance Stores",
  "Furniture Stores",
  "Home Furnishings Stores",
  "Specialty Food Stores",
  "Sporting Goods, Hobby, and Musical Instrument Stores",
  "Book Stores and News Dealers"
)

entertainment_recreation = c(
  "Other Amusement and Recreation Industries",
  "Museums, Historical Sites, and Similar Institutions",
  "Promoters of Performing Arts, Sports, and Similar Events",
  "Radio and Television Broadcasting",
  "Sound Recording Industries"
)

personal_services = c(
  "Personal Care Services",
  "Drycleaning and Laundry Services",
  "Other Personal Services",
  "Personal and Household Goods Repair and Maintenance"
)

hospitality_lodging = c(
  "Traveler Accommodation",
  "Bed and Breakfast Inns",
  "Resorts",
  "Extended Stay Hotels"
)

office_professional = c(
  "Accounting, Tax Preparation, Bookkeeping, and Payroll Services",
  "Legal Services",
  "Architectural, Engineering, and Related Services",
  "Agencies, Brokerages, and Other Insurance Related Activities",
  "Offices of Physicians",
  "Offices of Dentists",
  "Offices of Other Health Practitioners",
  "Real Estate Agencies"
)
```

```
target_categories = c("Drinking Places (Alcoholic Beverages)", "Restaurants and Other Eating Places", "

df_long_model_filtered_1 = df_long |>
  mutate(non_restaurant = if_else(top_category %in% target_categories, "No", "Yes"))
```

Exploratory Data Analysis (EDA)

Zip Code Flow Matrix

```
# Remove rows with NA in visitor_zip or poi_zip
zip_matrix = final_df |>
  filter(!is.na(visitor_zip) & !is.na(poi_zip)) |>
  group_by(visitor_zip, poi_zip) |>
  summarize(total_visitors = sum(visitors, na.rm = TRUE)) |>
  pivot_wider(names_from = poi_zip, values_from = total_visitors, values_fill = 0)
```

```
## `summarise()` has grouped output by 'visitor_zip'. You can override using the
## `.groups` argument.
```

```
# Convert to matrix and plot
zip_matrix_plot = zip_matrix |>
  column_to_rownames("visitor_zip") |>
  as.matrix() |>
  heatmap(
    col = colorRampPalette(c("white", "red"))(100),
    scale = "none",
    main = "Zip-to-Zip Visitor Flow",
    xlab = "Destination ZIP (To)",
    ylab = "Origin ZIP (From)"
  )
```

State to Destination ZIP in NYC

```
final_df_with_state = final_df |>
  left_join(df_tract_zip, by = c("visitor_zip" = "zip")) |>
  select(!zip) |>
  rename(visitor_state = usps_zip_pref_state)
```

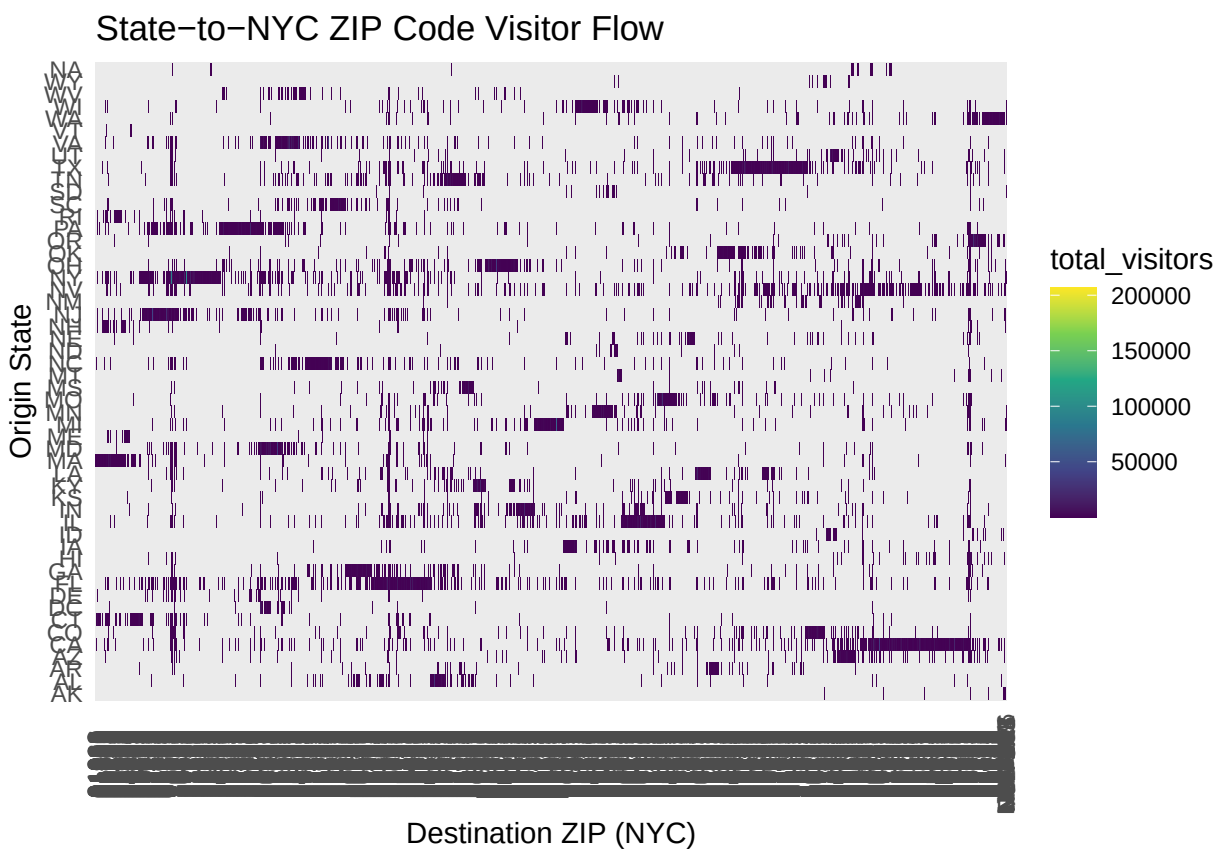
```
## Warning in left_join(final_df, df_tract_zip, by = c(visitor_zip = "zip")): Detected an unexpected ma
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 103268 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
state_zip_matrix = final_df_with_state |>
  group_by(visitor_state, poi_zip) |>
  summarize(total_visitors = sum(visitors, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'visitor_state'. You can override using the
## `.groups` argument.
```

```
ggplot(state_zip_matrix, aes(x = poi_zip, y = visitor_state, fill = total_visitors)) +
  geom_tile() +
  scale_fill_viridis_c() +
```

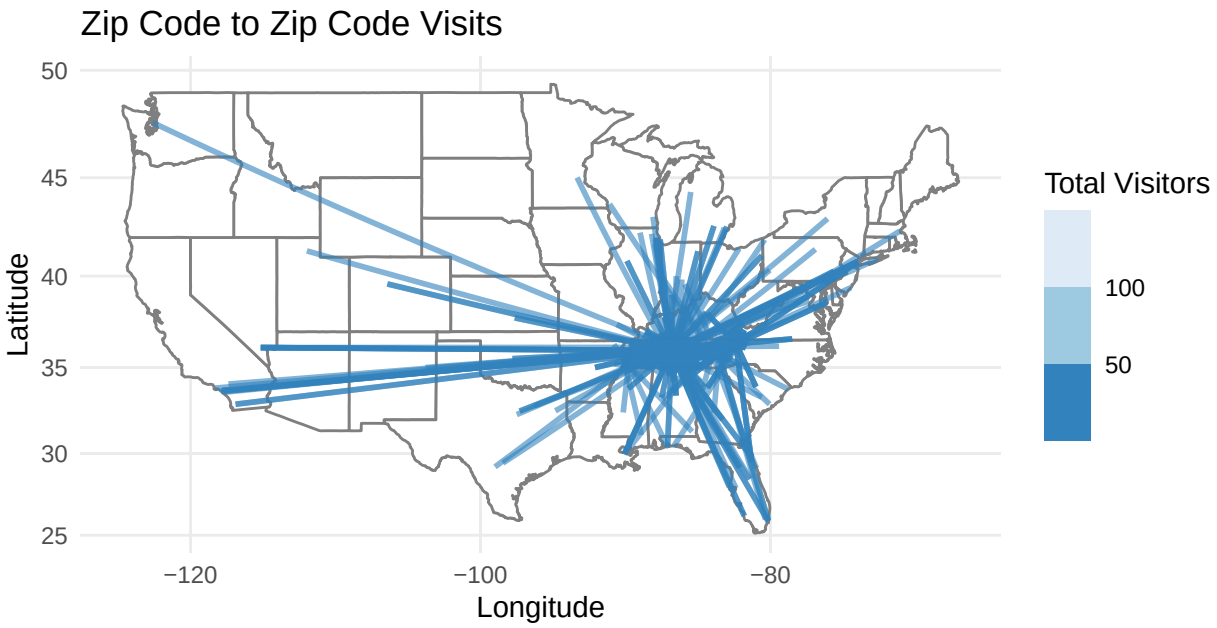
```
  labs(
    title = "State-to-NYC ZIP Code Visitor Flow",
    x = "Destination ZIP (NYC)",
    y = "Origin State"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```




```
nyc_zip_visitors = final_df |>
  group_by(poi_long, poi_lat) |>
  summarize(total_visitors = sum(visitors, na.rm = TRUE))


ggplot(nyc_zip_visitors, aes(x = poi_long, y = poi_lat)) +
  stat_density_2d(aes(fill = after_stat(level)), geom = "polygon", color = NA) +
  scale_fill_viridis_c() +
  labs(
    title = "Heatmap of NYC POIs by Visitor Counts",
    x = "Longitude",
    y = "Latitude"
  ) +
  theme_minimal()
```

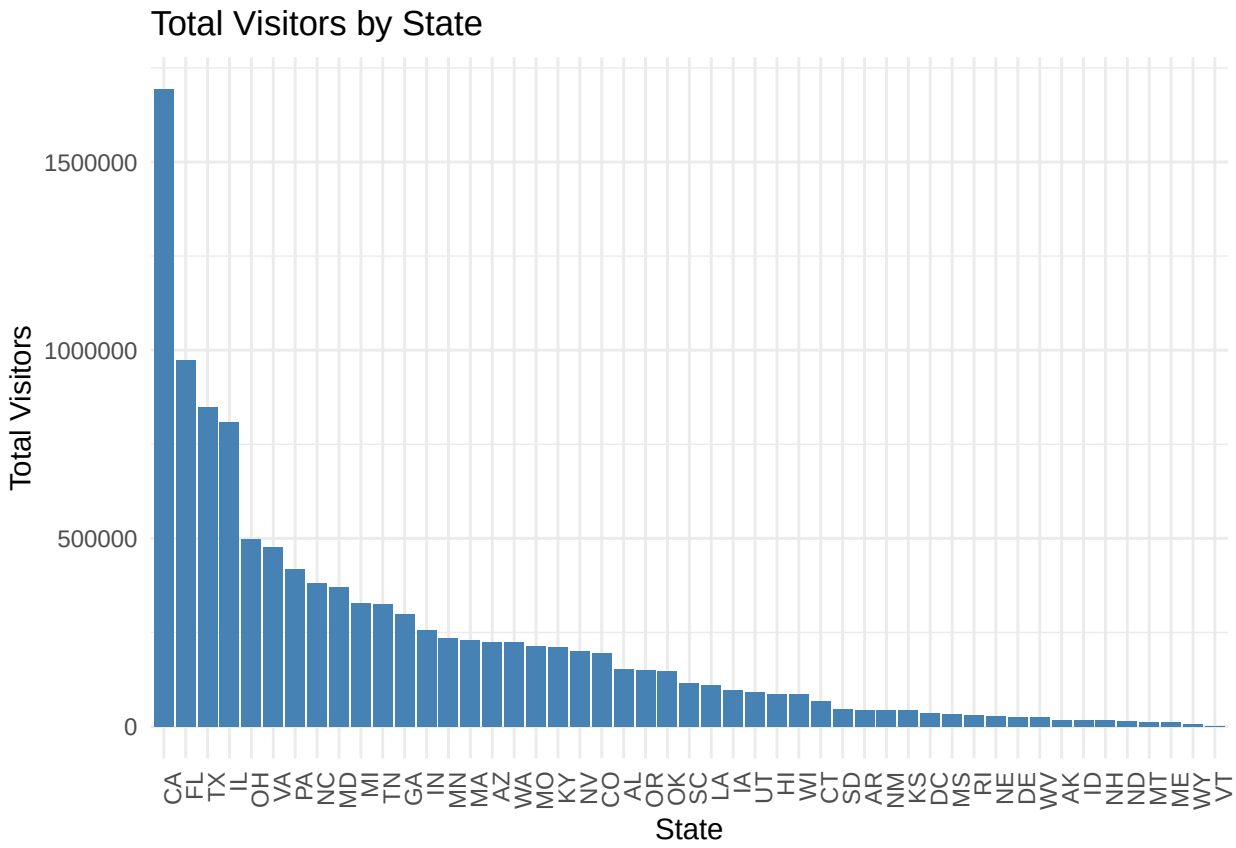
Flow map - work in progress

```r
#excise visitors from hawaii, include the second filter argument to get a better look at lower ends of
flow_df = final_df |>
  filter(vis_long > -150) |>
  filter(poi_long > -150) |>
  filter(vis_lat < 60) |>
  filter(poi_lat < 50) |>
  left_join(df_tract_zip, by = c("visitor_zip" = "zip"), multiple = "any") |>
  select(!zip) |>
  rename(visitor_state = usps_zip_pref_state) |>
  left_join(df_tract_zip, by = c("poi_zip" = "zip"), multiple = "any") |>
  rename(poi_state = usps_zip_pref_state) |> filter(poi_state == "TN")

# if you want to include visitors from hawaii
# flow_df = final_df

usa = map_data("state")

usa = rename(usa, state = "region")

usa$state = str_to_title(usa$state)

stateData = usa |>
  arrange(state, group, order)

ggplot() +
  geom_polygon(data = stateData,
               aes(x = long, y = lat, group = group),
               fill = "white", color = "gray50") +

  geom_segment(data = flow_df,
               aes(x = vis_long, y = vis_lat,
                   xend = poi_long, yend = poi_lat,
                   color = total_visitors),
               alpha = 0.6, linewidth = 1) +
  scale_color_fermenter(name = "Total Visitors", direction = -1) +
  coord_map() +
  theme_minimal() +
  labs(color = "Volume of Visits",
       title = "Zip Code to Zip Code Visits",
       x = "Longitude",
       y = "Latitude")
```



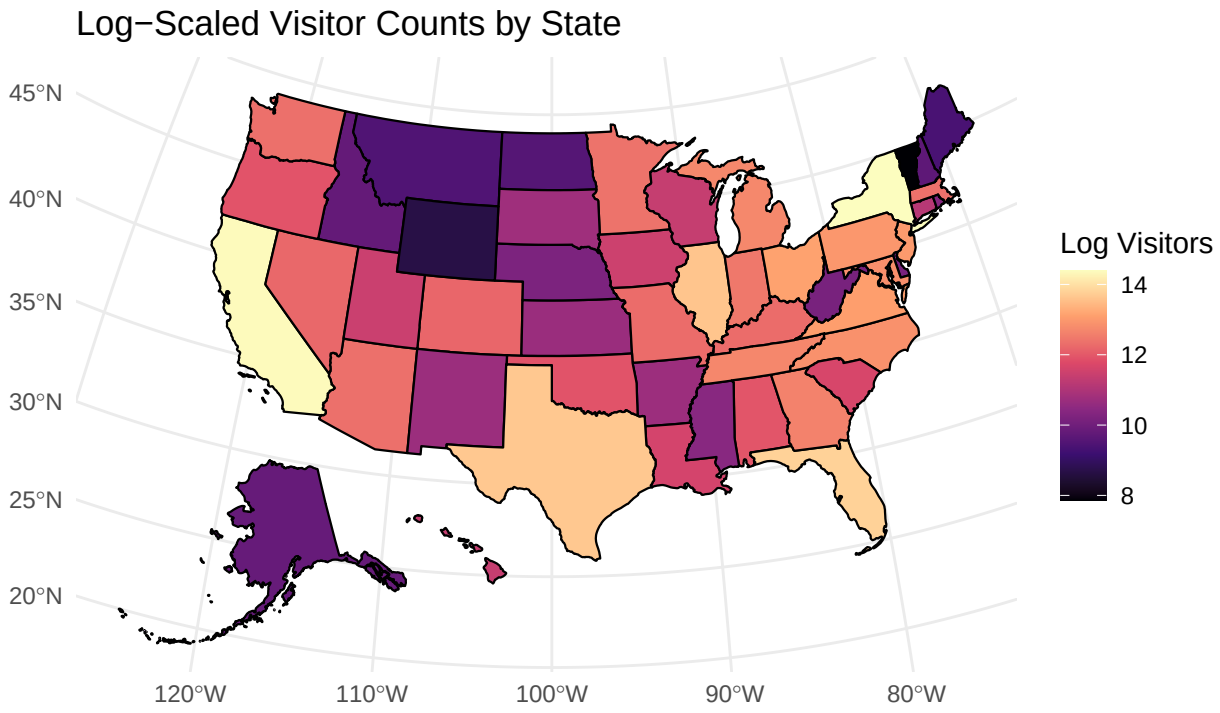
Visitor counts

```
state_visitors = final_df_with_state |>
  filter(visitor_state != "NY") |>
  filter(visitor_state != "NJ") |>
  group_by(visitor_state) |>
  summarize(total_visitors = sum(visitors, na.rm = TRUE)) |>
  arrange(desc(total_visitors))

ggplot(state_visitors, aes(x = reorder(visitor_state, -total_visitors), y = total_visitors)) +
  geom_col(fill = "steelblue") +
  labs(
    title = "Total Visitors by State",
    x = "State",
    y = "Total Visitors"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

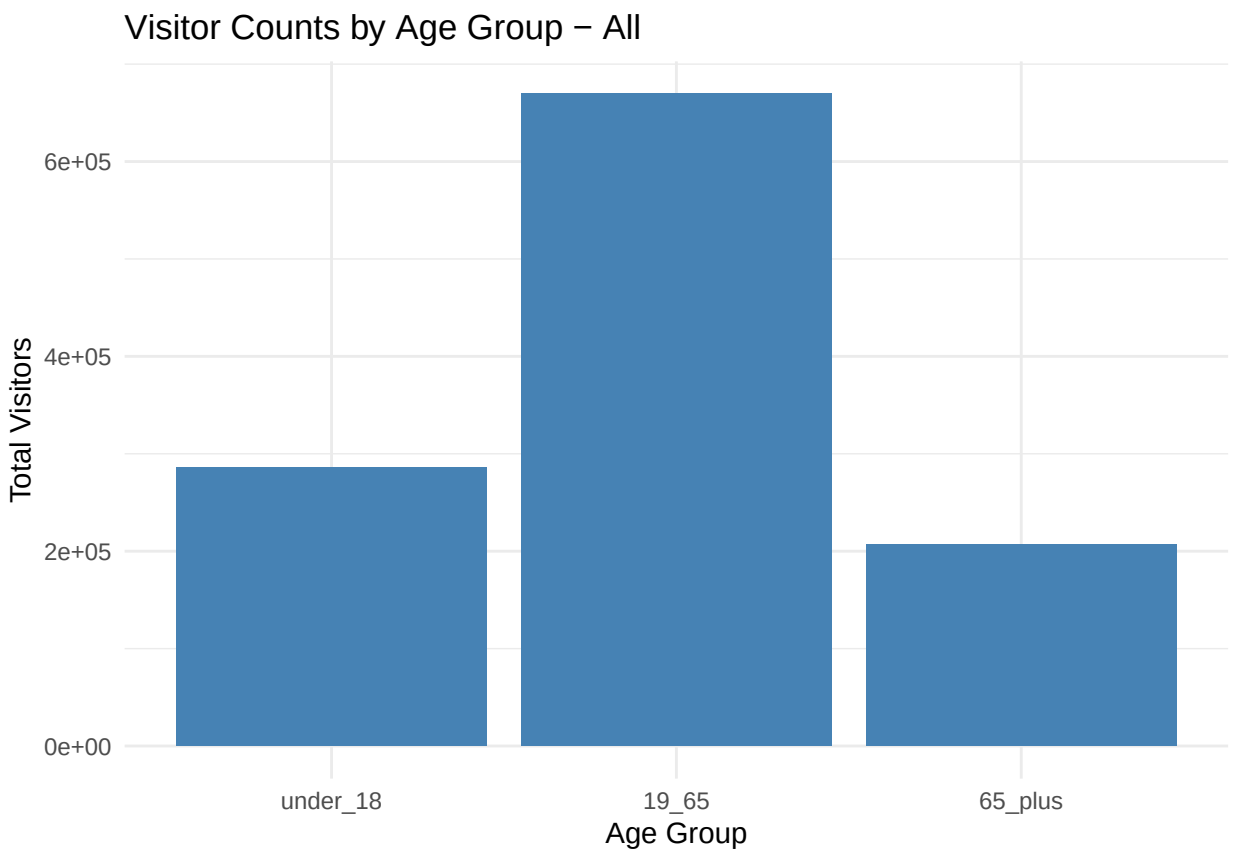


Map view of visitor counts

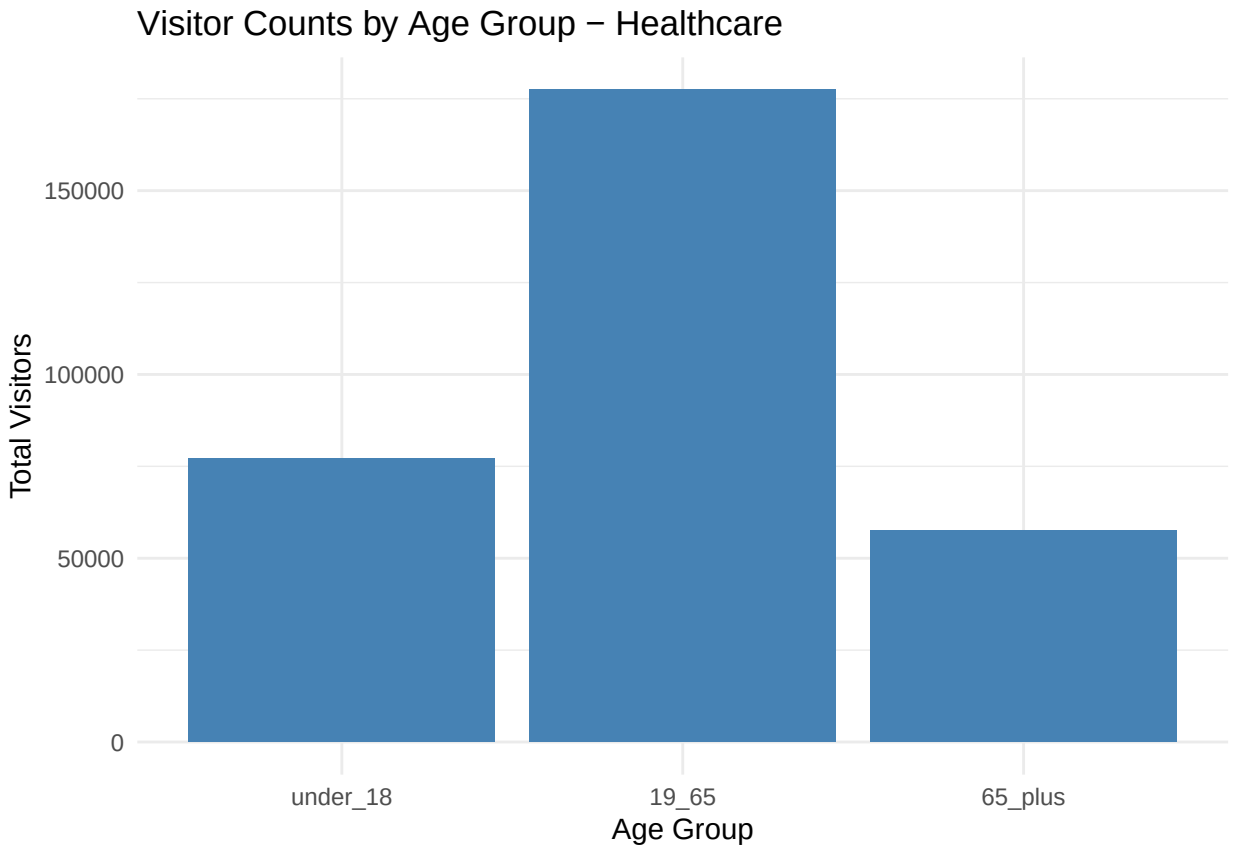
```
# Aggregate visitor counts by state
state_visitors_map = final_df_with_state |>
  group_by(visitor_state) |>
  summarize(total_visitors = sum(visitors, na.rm = TRUE)) |>
  filter(!is.na(visitor_state)) |>
  mutate(log_visitors = log1p(total_visitors)) |>
  rename(state = visitor_state)

# Plot using log scale
plot_usmap(data = state_visitors_map, regions = "states", values = "log_visitors") +
  scale_fill_viridis_c(name = "Log Visitors", option = "magma") +
  labs(title = "Log-Scaled Visitor Counts by State") +
  theme_minimal()
```



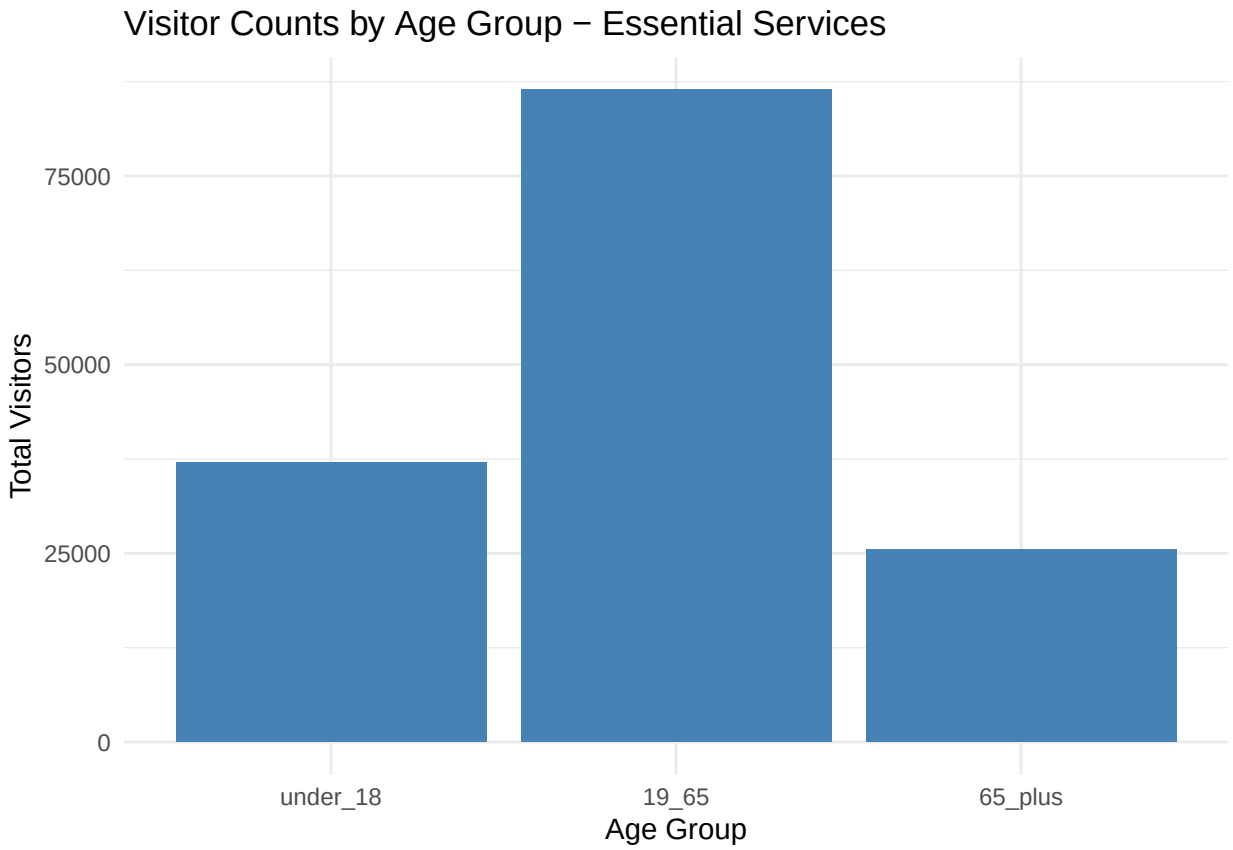

```
# Function to create bar plots for each category
plot_age_group_counts = function(df, category_name, category_vector) {
  df_filtered = df |>
    filter(top_category %in% category_vector) |>
    group_by(age_group) |>
    summarize(total_visitors = sum(visitor_count, na.rm = TRUE), .groups = "drop")

  ggplot(df_filtered, aes(x = age_group, y = total_visitors)) +
    geom_col(fill = "steelblue") +
    labs(
      title = paste("Visitor Counts by Age Group -", category_name),
      x = "Age Group",
      y = "Total Visitors"
    ) +
    theme_minimal() +
    theme(
      axis.text.x = element_text(angle = 0, hjust = 0.5),
      legend.position = "none"  # Remove legend
    )
}

# Generate bar plots for each category
plot_age_group_counts(df_long, "All", all_cat)
```



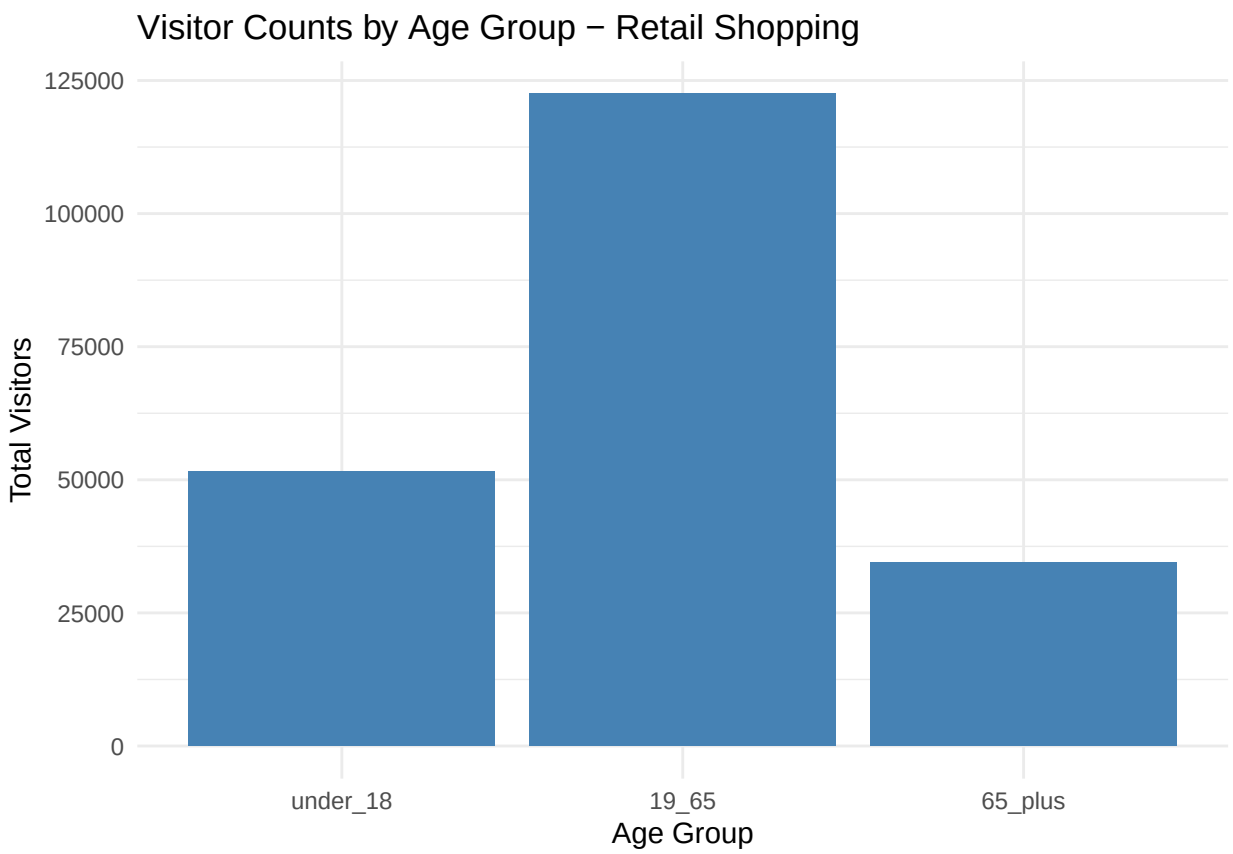

```
plot_age_group_counts(df_long, "Healthcare", medical_services)
```



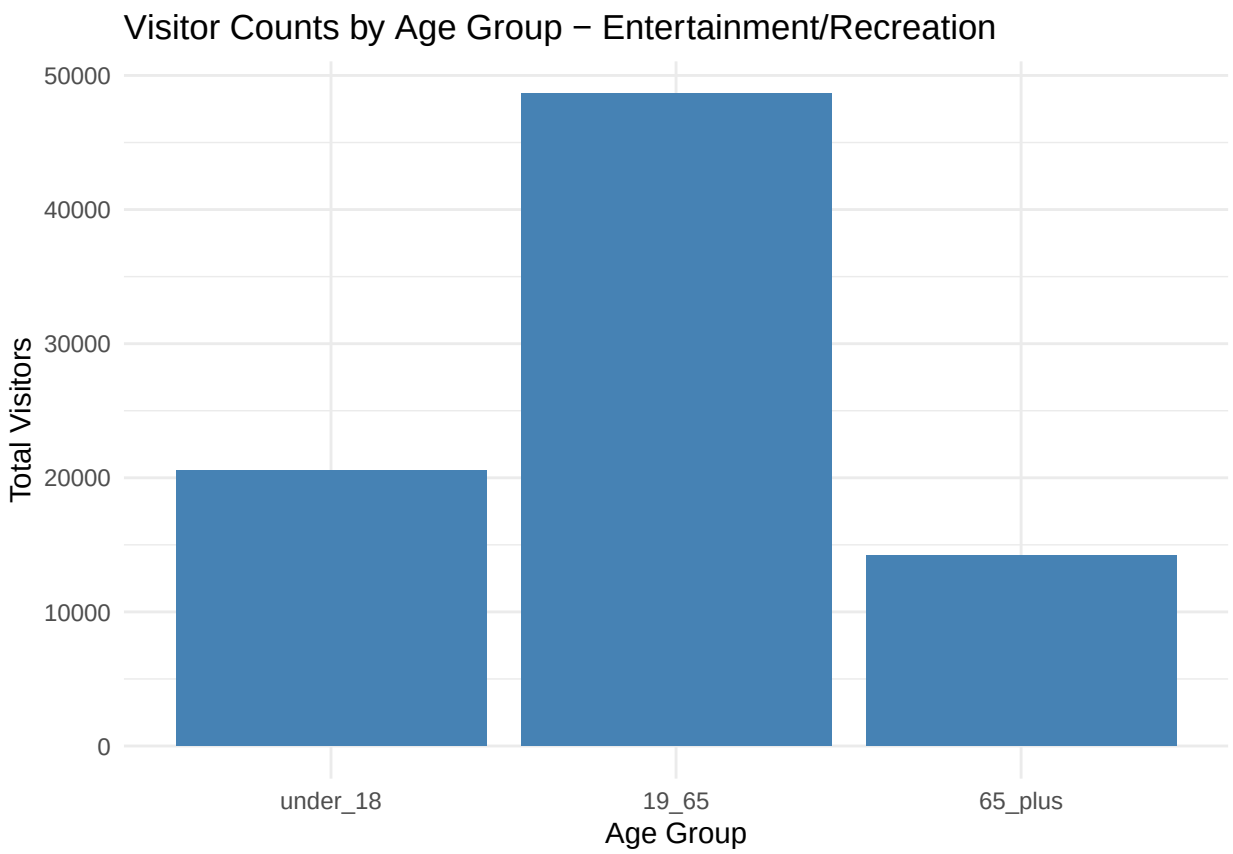

```
plot_age_group_counts(df_long, "Essential Services", essential_services)
```



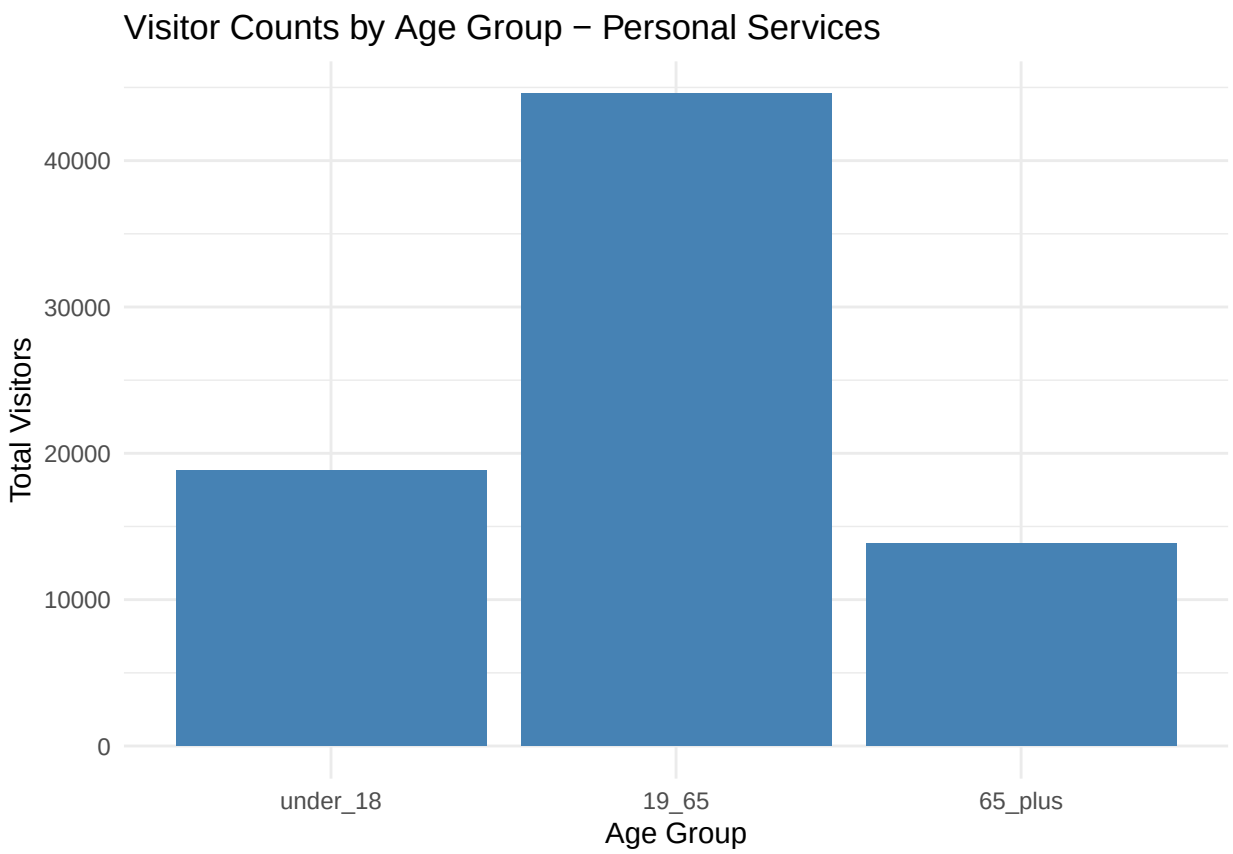

```
plot_age_group_counts(df_long, "Retail Shopping", retail_shopping)
```



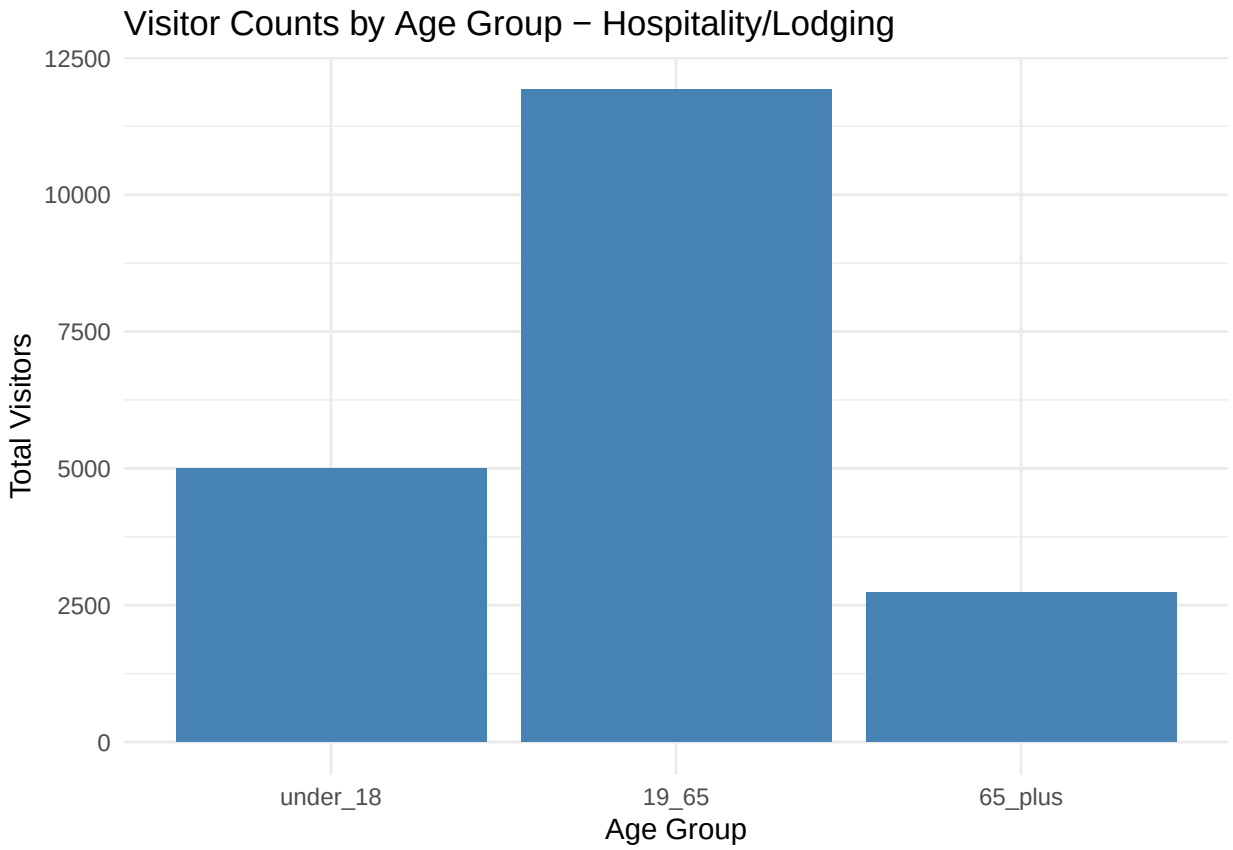

```
plot_age_group_counts(df_long, "Entertainment/Recreation", entertainment_recreation)
```



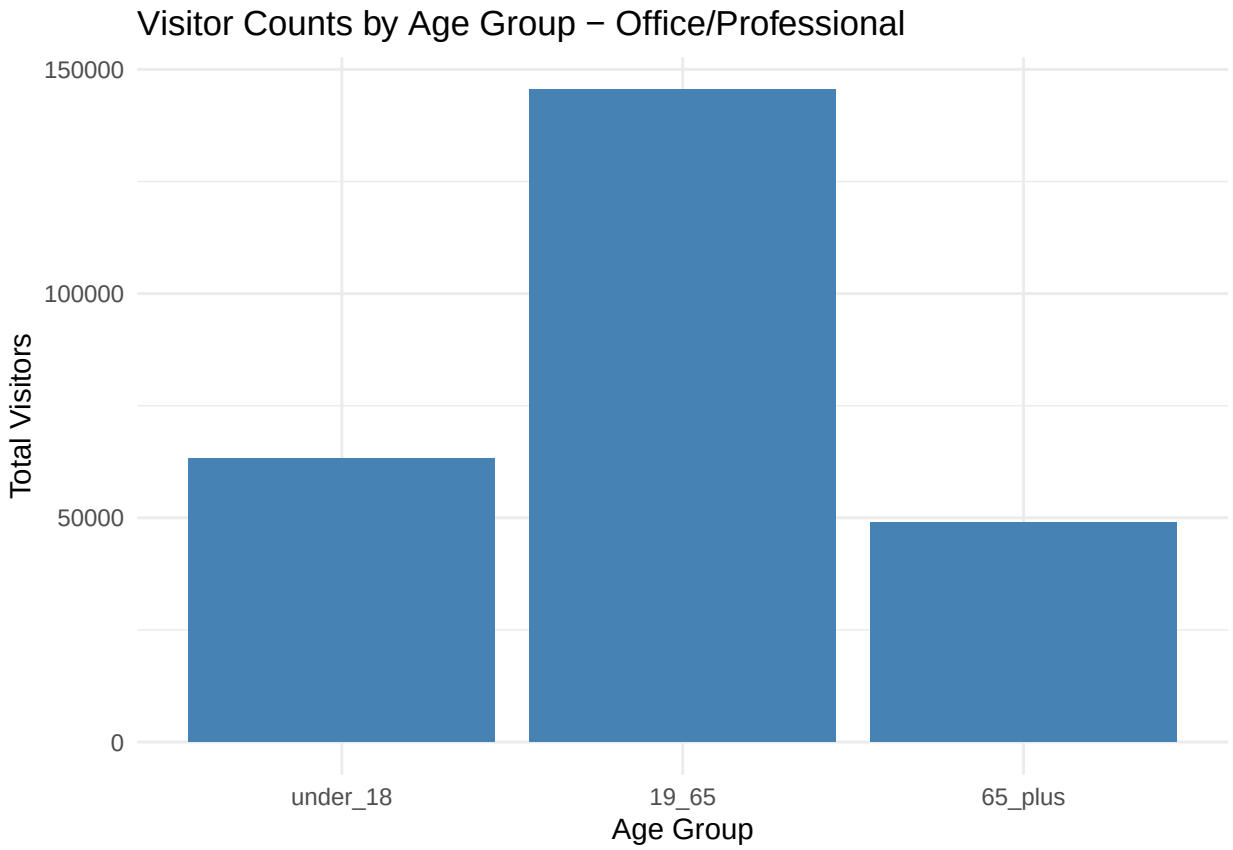

```
plot_age_group_counts(df_long, "Personal Services", personal_services)
```



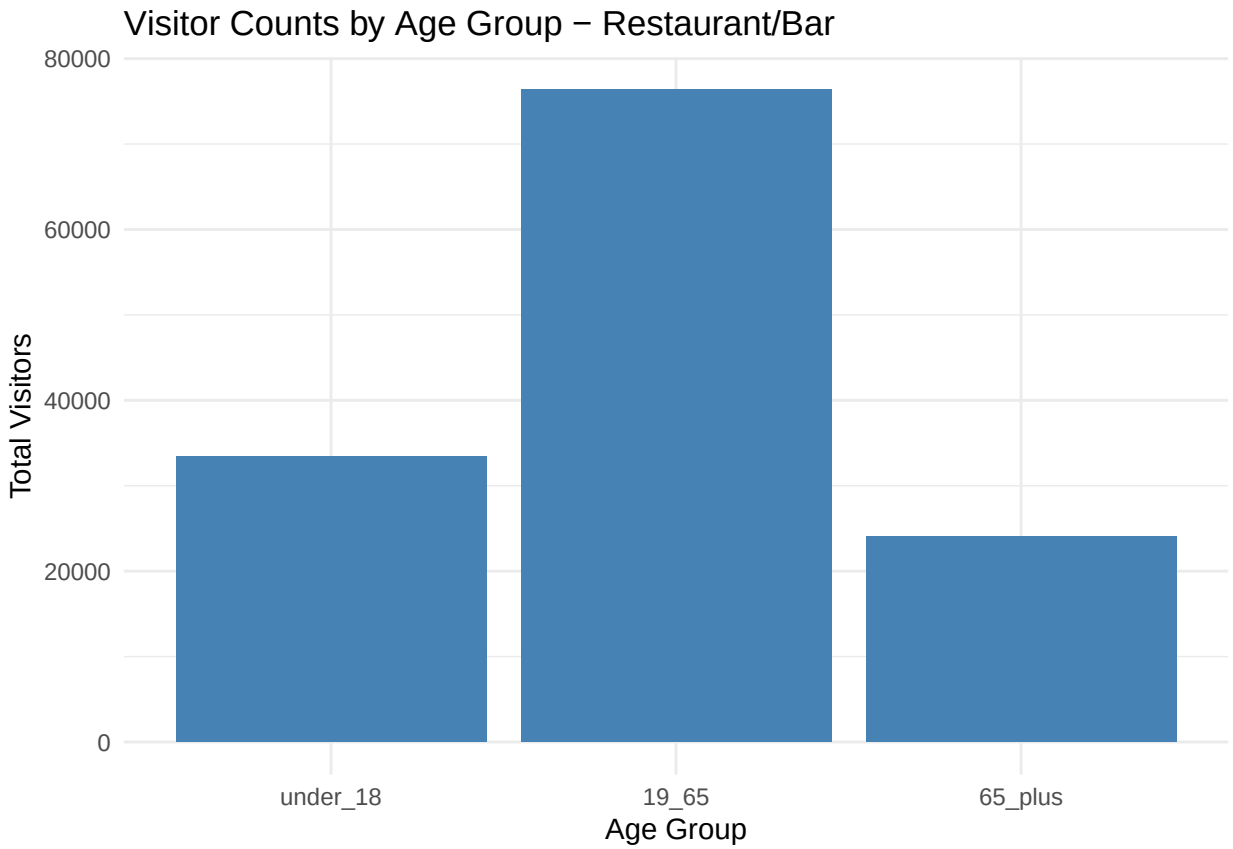

```
plot_age_group_counts(df_long, "Hospitality/Lodging", hospitality_lodging)
```




```
plot_age_group_counts(df_long, "Office/Professional", office_professional)
```

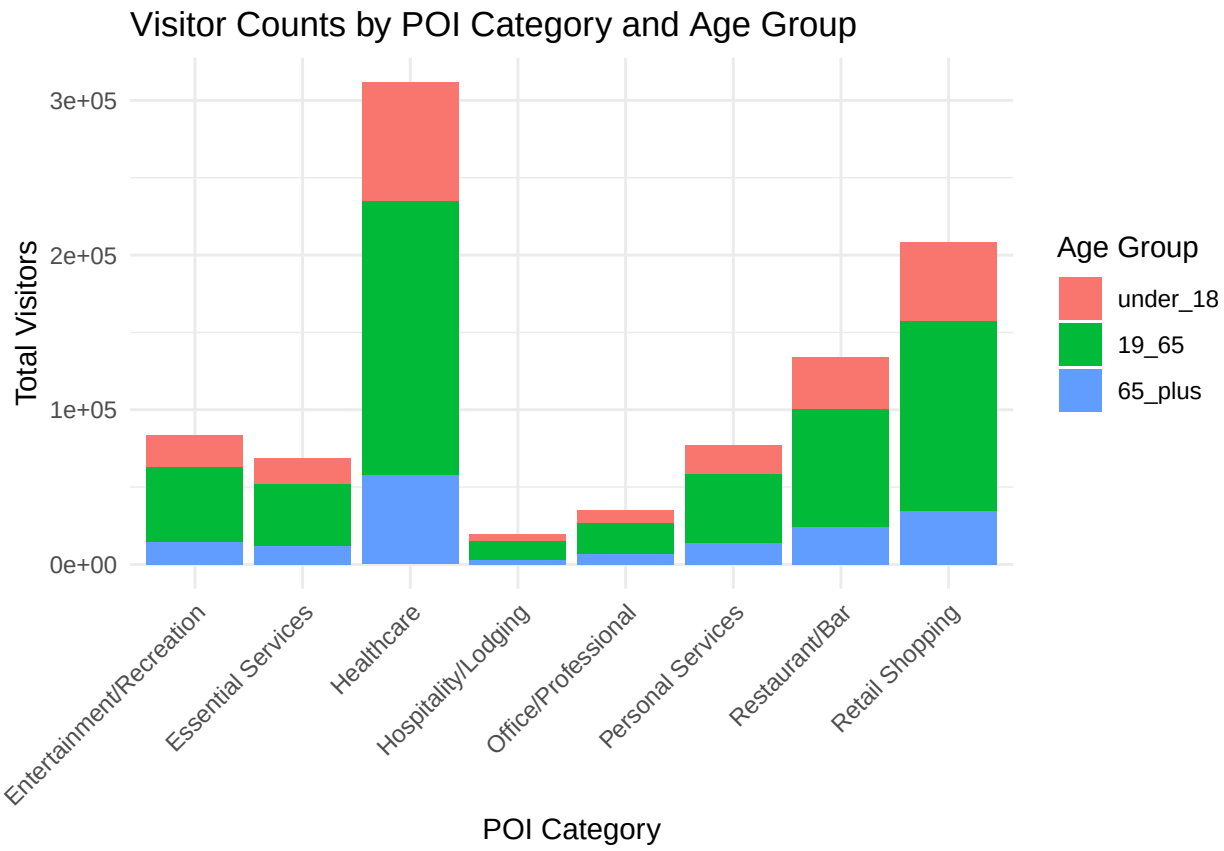



```
plot_age_group_counts(df_long, "Restaurant/Bar", target_categories)
```



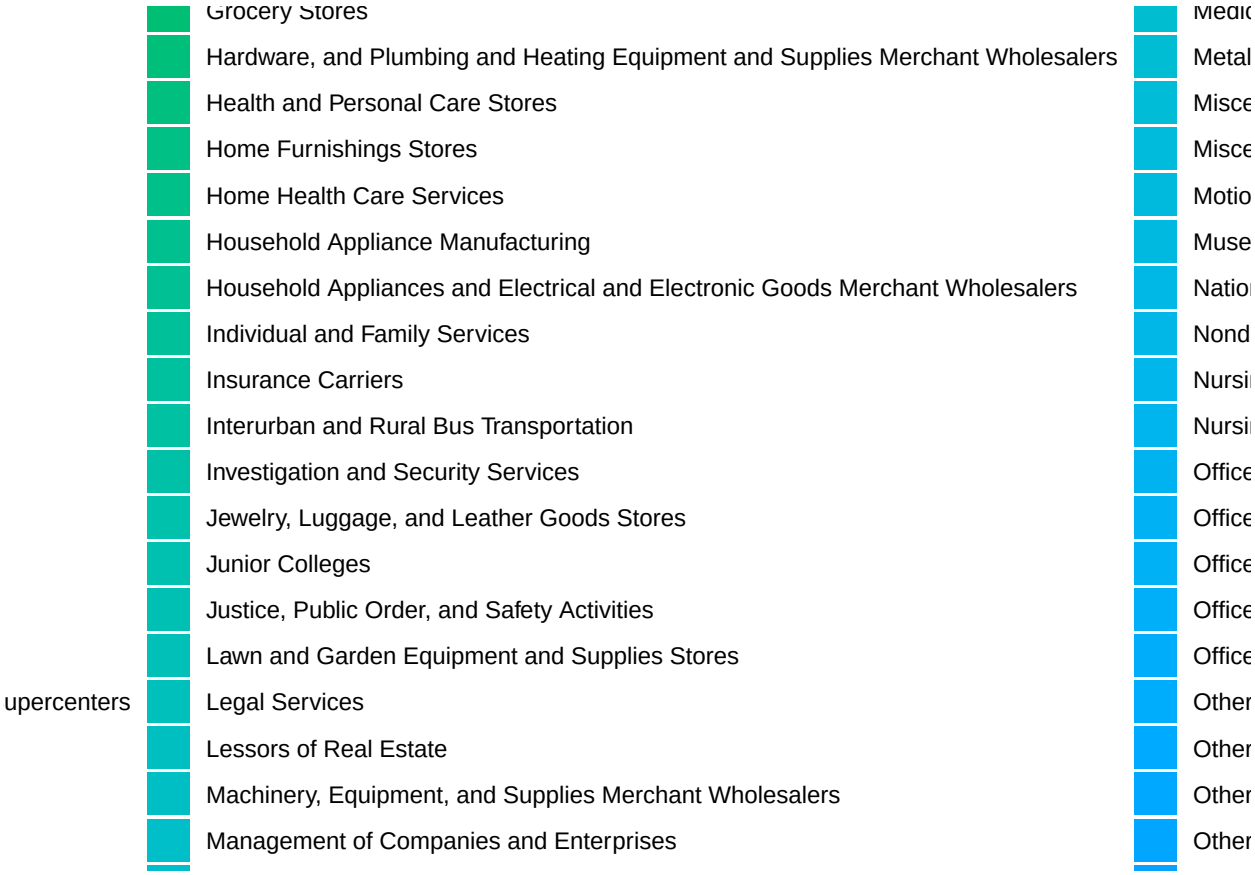

```
# Aggregate visitor counts by age group and POI category
df_filtered = df_long |>
  mutate(category_group = case_when(
    top_category %in% medical_services ~ "Healthcare",
    top_category %in% essential_services ~ "Essential Services",
    top_category %in% retail_shopping ~ "Retail Shopping",
    top_category %in% entertainment_recreation ~ "Entertainment/Recreation",
    top_category %in% personal_services ~ "Personal Services",
    top_category %in% hospitality_lodging ~ "Hospitality/Lodging",
    top_category %in% office_professional ~ "Office/Professional",
    top_category %in% target_categories ~ "Restaurant/Bar",
    TRUE ~ "Other"
  )) |>
  filter(category_group != "Other") |>  # Exclude any unintended categories
  group_by(category_group, age_group) |>
  summarize(total_visitors = sum(visitor_count, na.rm = TRUE), .groups = "drop")

# Create stacked bar plot
ggplot(df_filtered, aes(x = category_group, y = total_visitors, fill = age_group)) +
  geom_col(position = "stack") +
  labs(
    title = "Visitor Counts by POI Category and Age Group",
    x = "POI Category",
    y = "Total Visitors",
    fill = "Age Group"
  ) +
  theme_minimal() +
```

```
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),  # Rotate x-axis labels for readability
    legend.position = "right"  # Keep legend for age group colors
  )
```



Bar Plot

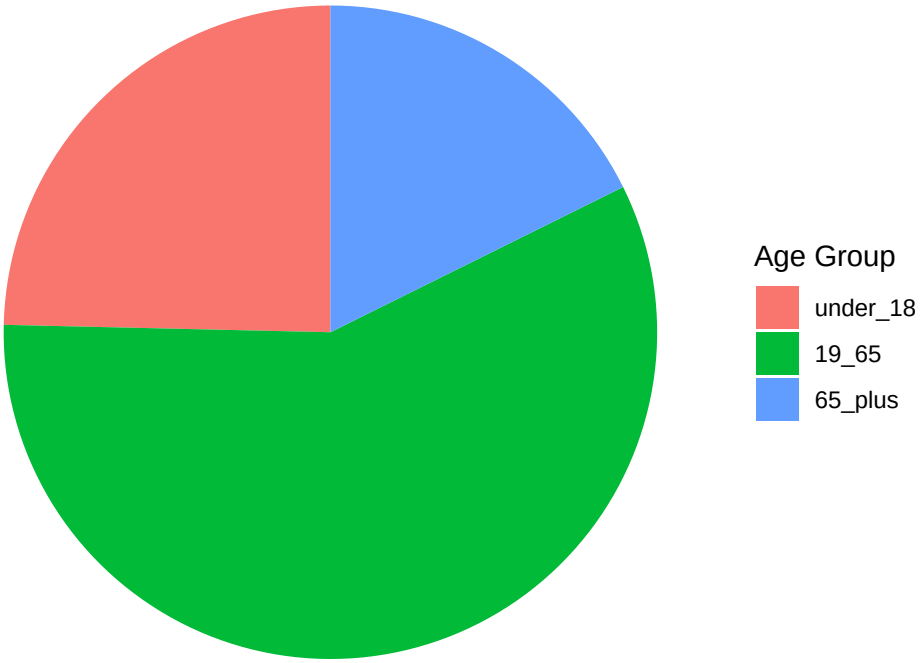
```
age_group_summary = df_long_model_filtered_1 |>
  group_by(age_group, top_category) |>
  summarize(total_visitors = sum(visitor_count), .groups = "drop")

ggplot(age_group_summary, aes(x = age_group, y = total_visitors, fill = top_category)) +
  geom_col(position = "dodge") +
  labs(
    title = "Visitor Counts by Age Group and Location Type",
    x = "Age Group",
    y = "Total Visitors",
    fill = "Location Type"
  ) +
  theme_minimal()
```



Pie chart (alternative to above)

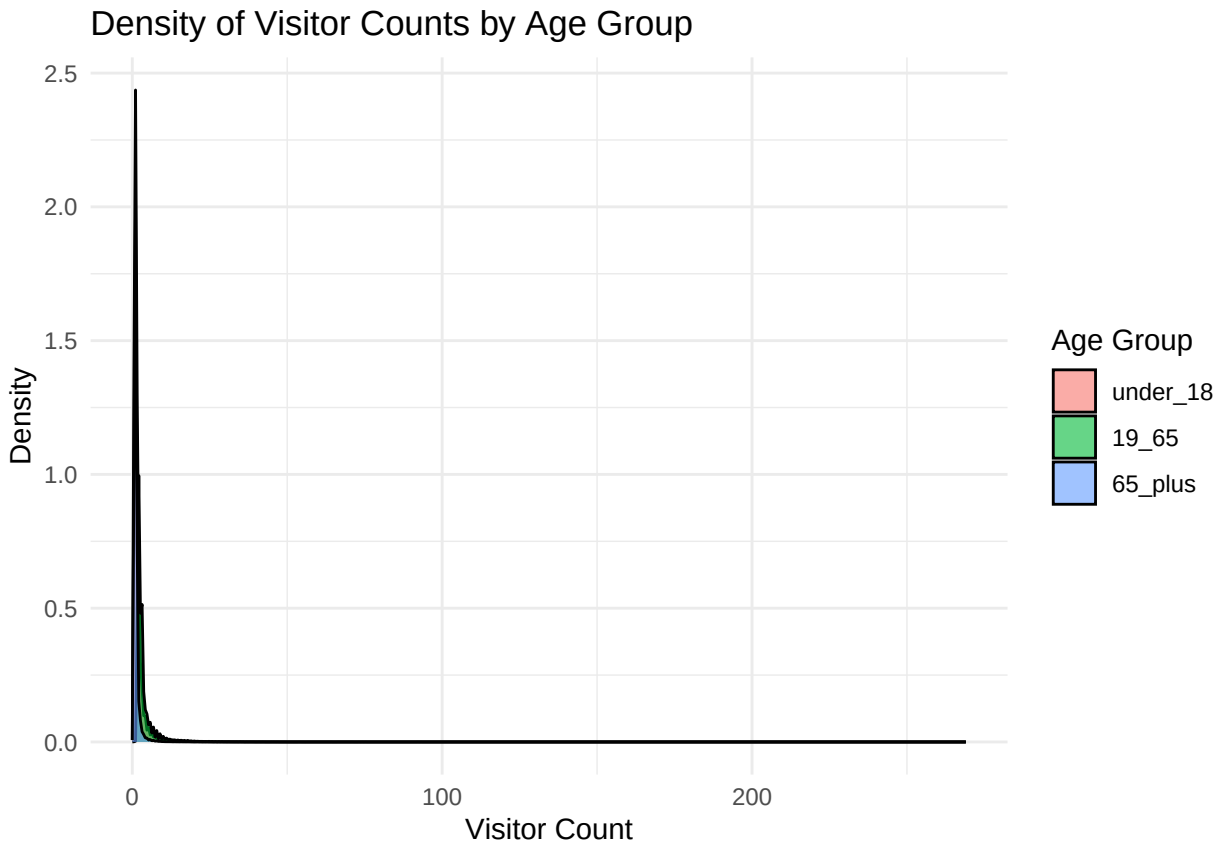
```
age_group_proportions = df_long_model_filtered_1 |>
  group_by(age_group) |>
  summarize(total_visitors = sum(visitor_count), .groups = "drop") |>
  mutate(proportion = total_visitors / sum(total_visitors))

ggplot(age_group_proportions, aes(x = "", y = proportion, fill = age_group)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  labs(
    title = "Proportion of Visitors by Age Group",
    fill = "Age Group"
  ) +
  theme_void()
```

Proportion of Visitors by Age Group



Density Plot

```
ggplot(df_long_model_filtered_1, aes(x = visitor_count, fill = age_group)) +
  geom_density(alpha = 0.6) +
  labs(
    title = "Density of Visitor Counts by Age Group",
    x = "Visitor Count",
    y = "Density",
    fill = "Age Group"
  ) +
  theme_minimal()
```



Modeling

All categories vs. categories of interest

```
df_long_model_filtered_1 = df_long |>
  mutate(non_restaurant = if_else(top_category %in% target_categories, "No", "Yes"))

poisson_model_interact_1 = glm(visitor_count ~ age_group * non_restaurant, family = poisson(link = "log
summary(poisson_model_interact_1)
```

```
##
## Call:
## glm(formula = visitor_count ~ age_group * non_restaurant, family = poisson(link = "log"),
##     data = df_long_model_filtered_1)
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        0.488807   0.005469  89.378  < 2e-16 ***
## age_group19_65                     0.825466   0.006558 125.866  < 2e-16 ***
## age_group65_plus                  -0.327818   0.008451 -38.790  < 2e-16 ***
## non_restaurantYes                  0.076497   0.005764  13.272  < 2e-16 ***
## age_group19_65:non_restaurantYes   0.028144   0.006909   4.074 4.63e-05 ***
## age_group65_plus:non_restaurantYes -0.005311   0.008908  -0.596    0.551
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1344355  on 576194  degrees of freedom
## Residual deviance:  989710  on 576189  degrees of freedom
## AIC: 2412303
##
## Number of Fisher Scoring iterations: 5
```

```
dispersion_test = sum(residuals(poisson_model_interact_1, type = "pearson")^2) / poisson_model_interact_
print(dispersion_test)
```

```
## [1] 4.421399
```

```
#Overdispersion present, use NB
```

NB models, overdispersion was present

NB model on whole data

"Are older individuals visiting restaurants/bars at lower rates compared to other age groups?" A negative estimate implies that an age group is visiting a location at a lower rate than the reference

```
nb_whole = glm.nb(visitor_count ~ age_group, data = df_long)
summary(nb_whole)
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group, data = df_long, init.theta = 3.465116764,
##     link = log)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.557409   0.002118   263.2   <2e-16 ***
## age_group19_65    0.850840   0.002694   315.8   <2e-16 ***
## age_group65_plus -0.332598   0.003185  -104.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(3.4651) family taken to be 1)
##
##     Null deviance: 608040  on 576194  degrees of freedom
## Residual deviance: 400689  on 576192  degrees of freedom
## AIC: 2089805
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  3.4651
##           Std. Err.:  0.0120
##
##  2 x log-likelihood:  -2089797.4910
```

```
df_model_filtered = df_long |>
  filter(top_category %in% target_categories)

nb_rest = glm.nb(visitor_count ~ age_group, data = df_model_filtered)
summary(nb_rest)
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group, data = df_model_filtered,
##     init.theta = 6.113489481, link = log)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.488807   0.006155   79.41   <2e-16 ***
## age_group19_65    0.825466   0.007679  107.50   <2e-16 ***
## age_group65_plus -0.327818   0.009347  -35.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(6.1135) family taken to be 1)
##
##     Null deviance: 62805  on 61520  degrees of freedom
## Residual deviance: 38746  on 61518  degrees of freedom
## AIC: 206801
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  6.1135
##           Std. Err.:  0.0906
##
##  2 x log-likelihood:  -206793.2540
```

NB with interaction

"Are older individuals visiting restaurants/bars at lower rates compared to other location types?"

```
run_nb_model = function(df, category_name, category_vector, reference_name, target_categories) {
  df_model = df |>
    filter(top_category %in% c(target_categories, category_vector)) |>  # Filter to only relevant POIs
    mutate(category_indicator = if_else(top_category %in% target_categories, reference_name, category_na
           category_indicator = factor(category_indicator, levels = c(reference_name, category_name)),
           age_group = factor(age_group, levels = c("under_18", "19_65", "65_plus")))  # Ensure Restaur

  nb_model = glm.nb(visitor_count ~ age_group * category_indicator + offset(log(total_visitors)), data =

  print(summary(nb_model))

  return(nb_model)
}


# All groups v. restaurant and bar
nb_model_1 = run_nb_model(df_long, "Non-Restaurant", all_cat, "Restaurant/Bar", target_categories)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

## Warning in glm.nb(visitor_count ~ age_group * category_indicator +
## offset(log(total_visitors)), : alternation limit reached

##
## Call:
## glm.nb(formula = visitor_count ~ age_group * category_indicator +
##     offset(log(total_visitors)), data = df_model, init.theta = 260176.8817,
##     link = log)
```

```
##
## Coefficients:
##                                                 Estimate Std. Error  z value
## (Intercept)                                    -1.387154   0.005469 -253.639
## age_group19_65                                  0.825464   0.006558  125.865
## age_group65_plus                               -0.327816   0.008451  -38.789
## category_indicatorNon-Restaurant               -0.017292   0.005820   -2.971
## age_group19_65:category_indicatorNon-Restaurant  0.029025   0.006976    4.161
## age_group65_plus:category_indicatorNon-Restaurant  0.003592   0.008993    0.399
##                                                Pr(>|z|)
## (Intercept)                                     < 2e-16 ***
## age_group19_65                                  < 2e-16 ***
## age_group65_plus                                < 2e-16 ***
## category_indicatorNon-Restaurant                0.00297 **
## age_group19_65:category_indicatorNon-Restaurant  3.17e-05 ***
## age_group65_plus:category_indicatorNon-Restaurant  0.68953
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(260175.5) family taken to be 1)
##
##     Null deviance: 361144  on 503510  degrees of freedom
## Residual deviance:  61717  on 503505  degrees of freedom
## AIC: 1300489
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  260177
##           Std. Err.:  83811
## Warning while fitting theta: alternation limit reached
##
##  2 x log-likelihood:  -1300475
```

```
# Healthcare v. restaurant and bar
nb_model_2 = run_nb_model(df_long, "Healthcare", medical_services, "Restaurant/Bar", target_categories)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group * category_indicator +
##     offset(log(total_visitors)), data = df_model, init.theta = 291940.95,
##     link = log)
##
## Coefficients:
##                                              Estimate Std. Error  z value
## (Intercept)                                 -1.387154   0.005469 -253.639
## age_group19_65                               0.825464   0.006558  125.865
## age_group65_plus                            -0.327817   0.008451  -38.789
## category_indicatorHealthcare                -0.011552   0.006549   -1.764
## age_group19_65:category_indicatorHealthcare  0.008897   0.007851    1.133
```

```
## age_group65_plus:category_indicatorHealthcare  0.035598   0.010090    3.528
##                                                Pr(>|z|)
## (Intercept)                                     < 2e-16 ***
## age_group19_65                                  < 2e-16 ***
## age_group65_plus                                < 2e-16 ***
## category_indicatorHealthcare                   0.077775 .
## age_group19_65:category_indicatorHealthcare    0.257082
## age_group65_plus:category_indicatorHealthcare 0.000418 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(291941) family taken to be 1)
##
##     Null deviance: 130295  on 206258  degrees of freedom
## Residual deviance:  22115  on 206253  degrees of freedom
## AIC: 525004
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  291941
##           Std. Err.:  156110
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -524990.2
```

```
# Essential Services v. restaurant and bar
nb_model_3 = run_nb_model(df_long, "Essential Services", essential_services, "Restaurant/Bar", target_ca
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group * category_indicator +
##     offset(log(total_visitors)), data = df_model, init.theta = 266135.8722,
##     link = log)
##
## Coefficients:
##                                                    Estimate Std. Error
## (Intercept)                                       -1.387154   0.005469
## age_group19_65                                     0.825464   0.006558
## age_group65_plus                                  -0.327816   0.008451
## category_indicatorEssential Services              -0.005183   0.007546
## age_group19_65:category_indicatorEssential Services    0.022406   0.009034
## age_group65_plus:category_indicatorEssential Services -0.043874   0.011732
##                                                    z value Pr(>|z|)
## (Intercept)                                       -253.639  < 2e-16 ***
## age_group19_65                                     125.865  < 2e-16 ***
## age_group65_plus                                   -38.789  < 2e-16 ***
## category_indicatorEssential Services                -0.687 0.492165
## age_group19_65:category_indicatorEssential Services      2.480 0.013130 *
## age_group65_plus:category_indicatorEssential Services   -3.740 0.000184 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(266135.9) family taken to be 1)
##
##     Null deviance: 87307  on 119297  degrees of freedom
## Residual deviance: 14283  on 119292  degrees of freedom
## AIC: 310129
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  266136
##           Std. Err.:  166582
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -310115.2
```

```
# Retail shopping v. restaurant and bar
nb_model_4 = run_nb_model(df_long, "Retail Shopping", retail_shopping, "Restaurant/Bar", target_categor
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group * category_indicator +
##     offset(log(total_visitors)), data = df_model, init.theta = 246266.0799,
##     link = log)
##
## Coefficients:
##                                                  Estimate Std. Error
## (Intercept)                                     -1.387154   0.005469
## age_group19_65                                   0.825464   0.006558
## age_group65_plus                                -0.327816   0.008451
## category_indicatorRetail Shopping               -0.009851   0.007022
## age_group19_65:category_indicatorRetail Shopping    0.039508   0.008401
## age_group65_plus:category_indicatorRetail Shopping -0.075457   0.010947
##                                                  z value Pr(>|z|)
## (Intercept)                                     -253.639  < 2e-16 ***
## age_group19_65                                   125.865  < 2e-16 ***
## age_group65_plus                                 -38.789  < 2e-16 ***
## category_indicatorRetail Shopping                 -1.403    0.161
## age_group19_65:category_indicatorRetail Shopping      4.703 2.56e-06 ***
## age_group65_plus:category_indicatorRetail Shopping   -6.893 5.47e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(246266.1) family taken to be 1)
##
##     Null deviance: 110607  on 139661  degrees of freedom
## Residual deviance:  17983  on 139656  degrees of freedom
## AIC: 367885
```

```
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  246266
##           Std. Err.:  133696
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -367870.8
```

```r
# Entertainment/Recreation v. restaurant and bar
nb_model_5 = run_nb_model(df_long, "Entertainment/Recreation", entertainment_recreation, "Restaurant/Ba
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in glm.nb(visitor_count ~ age_group * category_indicator +
## offset(log(total_visitors)), : alternation limit reached
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group * category_indicator +
##     offset(log(total_visitors)), data = df_model, init.theta = 261633.0894,
##     link = log)
##
## Coefficients:
##                                                              Estimate
## (Intercept)                                                -1.387154
## age_group19_65                                              0.825464
## age_group65_plus                                           -0.327816
## category_indicatorEntertainment/Recreation                 -0.014567
## age_group19_65:category_indicatorEntertainment/Recreation   0.037106
## age_group65_plus:category_indicatorEntertainment/Recreation -0.039001
##                                                            Std. Error  z value
## (Intercept)                                                  0.005469 -253.639
## age_group19_65                                               0.006558  125.865
## age_group65_plus                                             0.008451  -38.789
## category_indicatorEntertainment/Recreation                   0.008868   -1.643
## age_group19_65:category_indicatorEntertainment/Recreation    0.010598    3.501
## age_group65_plus:category_indicatorEntertainment/Recreation  0.013802   -2.826
##                                                            Pr(>|z|)
## (Intercept)                                                 < 2e-16 ***
## age_group19_65                                              < 2e-16 ***
## age_group65_plus                                            < 2e-16 ***
## category_indicatorEntertainment/Recreation                 0.100459
## age_group19_65:category_indicatorEntertainment/Recreation  0.000463 ***
## age_group65_plus:category_indicatorEntertainment/Recreation 0.004716 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(261639.6) family taken to be 1)
##
##     Null deviance: 67182  on 93320  degrees of freedom
## Residual deviance: 11300  on 93315  degrees of freedom
## AIC: 241907
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  261633
```

```
##           Std. Err.:  187367
## Warning while fitting theta: alternation limit reached
##
##  2 x log-likelihood:  -241893.1
```

```
# Personal Services v. restaurant and bar
nb_model_6 = run_nb_model(df_long, "Personal Services", personal_services, "Restaurant/Bar", target_cat
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group * category_indicator +
##     offset(log(total_visitors)), data = df_model, init.theta = 269684.1265,
##     link = log)
##
## Coefficients:
##                                                    Estimate Std. Error
## (Intercept)                                       -1.387154   0.005469
## age_group19_65                                     0.825464   0.006558
## age_group65_plus                                  -0.327816   0.008451
## category_indicatorPersonal Services               -0.024211   0.009112
## age_group19_65:category_indicatorPersonal Services  0.035867   0.010889
## age_group65_plus:category_indicatorPersonal Services  0.020254   0.014028
##                                                    z value Pr(>|z|)
## (Intercept)                                       -253.639  < 2e-16 ***
## age_group19_65                                     125.865  < 2e-16 ***
## age_group65_plus                                   -38.789  < 2e-16 ***
## category_indicatorPersonal Services                 -2.657 0.007880 **
## age_group19_65:category_indicatorPersonal Services   3.294 0.000988 ***
## age_group65_plus:category_indicatorPersonal Services   1.444 0.148786
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(269684.1) family taken to be 1)
##
##     Null deviance: 64015  on 95615  degrees of freedom
## Residual deviance: 11110  on 95610  degrees of freedom
## AIC: 245204
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  269684
##           Std. Err.:  204376
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -245190.2
```

```
#Hospitality/Lodging v. restaurant and bar
nb_model_7 = run_nb_model(df_long, "Hospitality/Lodging", hospitality_lodging, "Restaurant/Bar", target_
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group * category_indicator +
##     offset(log(total_visitors)), data = df_model, init.theta = 238968.7405,
##     link = log)
##
## Coefficients:
##                                                      Estimate Std. Error
## (Intercept)                                         -1.387154   0.005469
## age_group19_65                                       0.825464   0.006558
## age_group65_plus                                    -0.327816   0.008451
## category_indicatorHospitality/Lodging                0.017172   0.015175
## age_group19_65:category_indicatorHospitality/Lodging    0.044602   0.018092
## age_group65_plus:category_indicatorHospitality/Lodging -0.273315   0.025245
##                                                      z value Pr(>|z|)
## (Intercept)                                         -253.639   <2e-16 ***
## age_group19_65                                       125.865   <2e-16 ***
## age_group65_plus                                     -38.789   <2e-16 ***
## category_indicatorHospitality/Lodging                  1.132   0.2578
## age_group19_65:category_indicatorHospitality/Lodging      2.465   0.0137 *
## age_group65_plus:category_indicatorHospitality/Lodging  -10.827   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(238968.7) family taken to be 1)
##
##     Null deviance: 48248.0  on 66959  degrees of freedom
## Residual deviance:  8491.1  on 66954  degrees of freedom
## AIC: 173233
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  238969
##           Std. Err.:  210405
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -173219.2
```

```
#Office/Professional v. restaurant and bar
nb_model_8 = run_nb_model(df_long, "Office/Professional", office_professional, "Restaurant/Bar", target_
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

## Warning in glm.nb(visitor_count ~ age_group * category_indicator +
## offset(log(total_visitors)), : alternation limit reached

##
## Call:
## glm.nb(formula = visitor_count ~ age_group * category_indicator +
##     offset(log(total_visitors)), data = df_model, init.theta = 304279.271,
##     link = log)
##
## Coefficients:
```

```
##                                                          Estimate Std. Error
## (Intercept)                                             -1.387154   0.005469
## age_group19_65                                           0.825464   0.006558
## age_group65_plus                                        -0.327817   0.008451
## category_indicatorOffice/Professional                   -0.017431   0.006763
## age_group19_65:category_indicatorOffice/Professional    0.007652   0.008106
## age_group65_plus:category_indicatorOffice/Professional  0.070802   0.010378
##                                                          z value Pr(>|z|)
## (Intercept)                                             -253.639  < 2e-16 ***
## age_group19_65                                           125.865  < 2e-16 ***
## age_group65_plus                                         -38.789  < 2e-16 ***
## category_indicatorOffice/Professional                     -2.578  0.00995 **
## age_group19_65:category_indicatorOffice/Professional       0.944  0.34521
## age_group65_plus:category_indicatorOffice/Professional     6.822 8.96e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(304277.9) family taken to be 1)
##
##     Null deviance: 112037  on 191063  degrees of freedom
## Residual deviance:  19483  on 191058  degrees of freedom
## AIC: 481210
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  304279
##           Std. Err.:  178115
## Warning while fitting theta: alternation limit reached
##
##  2 x log-likelihood:  -481196.5
```

```
extract_nb_results = function(model, category_name) {
  results = broom.mixed::tidy(model) |>
    filter(grepl("category_indicator", term)) |>
    mutate(category = category_name) |>
    relocate(category) |>
    mutate(significance = case_when(
      p.value < 0.001 ~ "***",
      p.value < 0.01  ~ "**",
      p.value < 0.05  ~ "*",
      TRUE ~ ""
    ))

  return(results)
}

nb_summary_table = bind_rows(
  extract_nb_results(nb_model_1, "Non-Restaurant"),
  extract_nb_results(nb_model_2, "Healthcare"),
  extract_nb_results(nb_model_3, "Essential Services"),
  extract_nb_results(nb_model_4, "Retail Shopping"),
  extract_nb_results(nb_model_5, "Entertainment/Recreation"),
  extract_nb_results(nb_model_6, "Personal Services"),
```

```
  extract_nb_results(nb_model_7, "Hospitality/Lodging"),
  extract_nb_results(nb_model_8, "Office/Professional")
)

knitr::kable(nb_summary_table)
```

category	term	estimate	std.error	statistic	p.value	significance
Non-Restaurant	category_indicatorNon-Restaurant	-0.0172919	0.0058201	-2.9710780	0.0029676	**
Non-Restaurant	age_group19_65:category_indicatorNon-Restaurant	0.0290254	0.0069758	4.1608815	0.0000317	***
Non-Restaurant	age_group65_plus:category_indicatorNon-Restaurant	0.0035925	0.0089926	0.3994917	0.6895310	
Healthcare	category_indicatorHealthcare	-0.0115516	0.0065495	-1.7637418	0.0777755	
Healthcare	age_group19_65:category_indicatorHealthcare	0.0088974	0.0078508	1.1333157	0.2570817	
Healthcare	age_group65_plus:category_indicatorHealthcare	0.0359983	0.0100897	3.5281962	0.0004184	***
Essential Services	category_indicatorEssential Services	-0.0051828	0.0075456	-0.6868696	0.4921649	
Essential Services	age_group19_65:category_indicatorEssential Services	0.0224063	0.0090340	2.4802229	0.0131300	*
Essential Services	age_group65_plus:category_indicatorEssential Services	-0.0438742	0.0117320	-3.7396973	0.0001842	***
Retail Shopping	category_indicatorRetail Shopping	-0.0098513	0.0070216	-1.4029892	0.1606201	
Retail Shopping	age_group19_65:category_indicatorRetail Shopping	0.0395080	0.0084007	4.7029650	0.0000026	***
Retail Shopping	age_group65_plus:category_indicatorRetail Shopping	-0.0754574	0.0109473	-6.8928021	0.0000000	***
Entertainment/Recreation	category_indicatorEntertainment/Recreation	-0.0145671	0.0088681	-1.6426306	0.1004594	
Entertainment/Recreation	age_group19_65:category_indicatorEntertainment/Recreation	0.0371061	0.0105078	3.5012654	0.0004631	***
Entertainment/Recreation	age_group65_plus:category_indicatorEntertainment/Recreation	-0.0390010	0.0138617	-2.8258177	0.0047160	**
Personal Services	category_indicatorPersonal Services	-0.0242113	0.0091117	-2.6571591	0.0078802	**
Personal Services	age_group19_65:category_indicatorPersonal Services	0.0358668	0.0108890	3.2938478	0.0009883	***
Personal Services	age_group65_plus:category_indicatorPersonal Services	0.0202536	0.0140277	1.4438340	0.1487857	
Hospitality/Lodging	category_indicatorHospitality/Lodging	0.0171719	0.0151750	1.1315906	0.2578066	
Hospitality/Lodging	age_group19_65:category_indicatorHospitality/Lodging	0.0446250	0.0199222	2.2402852	0.0136904	*
Hospitality/Lodging	age_group65_plus:category_indicatorHospitality/Lodging	0.2733152	0.0252448	10.8265759	0.0000000	***
Office/Professional	category_indicatorOffice/Professional	-0.0174315	0.0067625	-2.5776629	0.0099471	**
Office/Professional	age_group19_65:category_indicatorOffice/Professional	0.0076516	0.0081062	0.9439125	0.3452144	
Office/Professional	age_group65_plus:category_indicatorOffice/Professional	0.0708256	0.0103780	6.8223588	0.0000000	***

NB no interaction and offset

```r
run_nb_model = function(df, category_name, category_vector) {
  df_model = df |>
    filter(top_category %in% category_vector) |>  # Filter to only relevant POIs
    mutate(
           age_group = factor(age_group, levels = c("under_18", "19_65", "65_plus")))  # Ensure age gro

  nb_model = glm.nb(visitor_count ~ age_group + offset(log(total_visitors)), data = df_model)

  print(summary(nb_model))

  return(nb_model)
}

# Run models for each POI type vs. Restaurant/Bar
nb_model_1 = run_nb_model(df_long, "Full", all_cat)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
```

```
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

## Warning in glm.nb(visitor_count ~ age_group + offset(log(total_visitors)), :
## alternation limit reached

##
## Call:
## glm.nb(formula = visitor_count ~ age_group + offset(log(total_visitors)),
##     data = df_model, init.theta = 260055.0825, link = log)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.402438   0.001871  -749.7   <2e-16 ***
## age_group19_65    0.851136   0.002235   380.9   <2e-16 ***
## age_group65_plus -0.324644   0.002888  -112.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(260049.6) family taken to be 1)
##
##     Null deviance: 361144  on 503510  degrees of freedom
## Residual deviance:  61739  on 503508  degrees of freedom
## AIC: 1300505
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  260055
##           Std. Err.:  83789
## Warning while fitting theta: alternation limit reached
##
##  2 x log-likelihood:  -1300497
```

```
nb_model_2 = run_nb_model(df_long, "Healthcare", medical_services)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

## Warning in glm.nb(visitor_count ~ age_group + offset(log(total_visitors)), :
## alternation limit reached

##
## Call:
## glm.nb(formula = visitor_count ~ age_group + offset(log(total_visitors)),
##     data = df_model, init.theta = 298129.6537, link = log)
##
```

```
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.398705   0.003604 -388.15   <2e-16 ***
## age_group19_65    0.834362   0.004315  193.34   <2e-16 ***
## age_group65_plus -0.292218   0.005512  -53.02   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(298132.6) family taken to be 1)
##
##     Null deviance: 90458  on 144737  degrees of freedom
## Residual deviance: 15252  on 144735  degrees of freedom
## AIC: 367472
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  298130
##           Std. Err.:  194662
## Warning while fitting theta: alternation limit reached
##
##  2 x log-likelihood:  -367463.7
```

```r
nb_model_3 = run_nb_model(df_long, "Essential Services", essential_services)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
```

```
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in glm.nb(visitor_count ~ age_group + offset(log(total_visitors)), :
## alternation limit reached
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group + offset(log(total_visitors)),
##     data = df_model, init.theta = 254314.7652, link = log)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.392336   0.005199 -267.83   <2e-16 ***
## age_group19_65    0.847870   0.006213  136.47   <2e-16 ***
## age_group65_plus -0.371690   0.008137  -45.68   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(254312.9) family taken to be 1)
##
##     Null deviance: 47469.5  on 57776  degrees of freedom
## Residual deviance:  7420.1  on 57774  degrees of freedom
## AIC: 152597
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  254315
##           Std. Err.:  214456
## Warning while fitting theta: alternation limit reached
##
##  2 x log-likelihood:  -152588.8
```

```r
nb_model_4 = run_nb_model(df_long, "Retail Shopping", retail_shopping)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group + offset(log(total_visitors)),
##     data = df_model, init.theta = 226721.0294, link = log)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.397005   0.004404 -317.23   <2e-16 ***
## age_group19_65    0.864971   0.005250  164.77   <2e-16 ***
## age_group65_plus -0.403273   0.006958  -57.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(226721) family taken to be 1)
##
##     Null deviance: 70769  on 78140  degrees of freedom
## Residual deviance: 11121  on 78138  degrees of freedom
## AIC: 210352
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  226721
##           Std. Err.:  150591
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -210344.4
```

```r
nb_model_5 = run_nb_model(df_long, "Entertainment/Recreation", entertainment_recreation)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group + offset(log(total_visitors)),
##     data = df_model, init.theta = 235021.6756, link = log)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.401721   0.006981 -200.79   <2e-16 ***
## age_group19_65    0.862570   0.008325  103.61   <2e-16 ***
## age_group65_plus -0.366817   0.010912  -33.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for Negative Binomial(235021.7) family taken to be 1)
##
##     Null deviance: 27344.7  on 31799  degrees of freedom
## Residual deviance:  4437.7  on 31797  degrees of freedom
## AIC: 84375
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  235022
##           Std. Err.:  262422
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -84366.61
```

```
nb_model_6 = run_nb_model(df_long, "Personal Services", personal_services)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group + offset(log(total_visitors)),
##     data = df_model, init.theta = 253207.0889, link = log)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.411365   0.007288 -193.66   <2e-16 ***
## age_group19_65    0.861330   0.008692   99.09   <2e-16 ***
## age_group65_plus -0.307563   0.011196  -27.47   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(253207.1) family taken to be 1)
##
##     Null deviance: 24177.6  on 34094  degrees of freedom
## Residual deviance:  4247.6  on 34092  degrees of freedom
## AIC: 87672
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  253207
##           Std. Err.:  331273
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -87663.67
```

```
nb_model_7 = run_nb_model(df_long, "Hospitality/Lodging", hospitality_lodging)
```

```
## Warning in glm.nb(visitor_count ~ age_group + offset(log(total_visitors)), :
## alternation limit reached
```

```
##
```

```
## Call:
## glm.nb(formula = visitor_count ~ age_group + offset(log(total_visitors)),
##     data = df_model, init.theta = 405.6820607, link = log)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.36936    0.01436  -95.36   <2e-16 ***
## age_group19_65     0.86412    0.01721   50.20   <2e-16 ***
## age_group65_plus -0.59647    0.02400  -24.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(405.6821) family taken to be 1)
##
##     Null deviance: 8006.6  on 5438  degrees of freedom
## Residual deviance: 1577.2  on 5436  degrees of freedom
## AIC: 15696
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  406
##           Std. Err.:  215
## Warning while fitting theta: alternation limit reached
##
##  2 x log-likelihood:  -15688.45
```

```
nb_model_8 = run_nb_model(df_long, "Office/Professional", office_professional)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in glm.nb(visitor_count ~ age_group + offset(log(total_visitors)), :
## alternation limit reached
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group + offset(log(total_visitors)),
##     data = df_model, init.theta = 318795.6055, link = log)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.404585   0.003978 -353.12   <2e-16 ***
## age_group19_65   0.833116   0.004764  174.86   <2e-16 ***
## age_group65_plus -0.257014   0.006023  -42.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(318794.5) family taken to be 1)
##
##     Null deviance: 72199  on 129542  degrees of freedom
## Residual deviance: 12620  on 129540  degrees of freedom
## AIC: 323678
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  318796
##           Std. Err.:  239191
## Warning while fitting theta: alternation limit reached
```

```
##
##  2 x log-likelihood:  -323670
```

```
nb_model_9 = run_nb_model(df_long, "Restaurant/Bar", target_categories)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
##
## Call:
## glm.nb(formula = visitor_count ~ age_group + offset(log(total_visitors)),
##     data = df_model, init.theta = 278617.7948, link = log)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.387154   0.005469 -253.64   <2e-16 ***
## age_group19_65    0.825464   0.006558  125.86   <2e-16 ***
## age_group65_plus -0.327817   0.008451  -38.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(278617.8) family taken to be 1)
##
##     Null deviance: 39837.6  on 61520  degrees of freedom
## Residual deviance:  6862.7  on 61518  degrees of freedom
## AIC: 157534
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  278618
##           Std. Err.:  259310
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -157526.5
```

```
extract_nb_results = function(model, category_name) {
  results = broom.mixed::tidy(model) |>
    mutate(significance = case_when(
      p.value < 0.001 ~ "***",
      p.value < 0.01  ~ "**",
      p.value < 0.05  ~ "*",
      TRUE ~ ""
    ))

  return(results)
}

# Combine results from all models
nb_summary_table = bind_rows(
  extract_nb_results(nb_model_1, "Non-Restaurant"),
  extract_nb_results(nb_model_2, "Healthcare"),
  extract_nb_results(nb_model_3, "Essential Services"),
```

```
  extract_nb_results(nb_model_4, "Retail Shopping"),
  extract_nb_results(nb_model_5, "Entertainment/Recreation"),
  extract_nb_results(nb_model_6, "Personal Services"),
  extract_nb_results(nb_model_7, "Hospitality/Lodging"),
  extract_nb_results(nb_model_8, "Office/Professional"),
  extract_nb_results(nb_model_9, "Restaurant/Bar")
)

nb_summary_table = nb_summary_table |>
  bind_cols(category = c("Full","Full","Full", "Healthcare", "Healthcare", "Healthcare", "Essential Ser
  relocate(category)

# Display as a table
knitr::kable(nb_summary_table)
```

category	term	estimate	std.error	statistic	p.value	significance
Full	(Intercept)	-1.4024376	0.0018707	-749.67650	0	***
Full	age_group19_65	0.8511363	0.0022347	380.87875	0	***
Full	age_group65_plus	-0.3246436	0.0028882	-112.40530	0	***
Healthcare	(Intercept)	-1.3987052	0.0036035	-388.14774	0	***
Healthcare	age_group19_65	0.8343616	0.0043155	193.34216	0	***
Healthcare	age_group65_plus	-0.2922183	0.0055116	-53.01844	0	***
Essential Services	(Intercept)	-1.3923364	0.0051986	-267.82935	0	***
Essential Services	age_group19_65	0.8478701	0.0062130	136.46705	0	***
Essential Services	age_group65_plus	-0.3716905	0.0081374	-45.67693	0	***
Retail Shopping	(Intercept)	-1.3970048	0.0044038	-317.22778	0	***
Retail Shopping	age_group19_65	0.8649715	0.0052497	164.76570	0	***
Retail Shopping	age_group65_plus	-0.4032735	0.0069584	-57.95473	0	***
Entertainment/Recreation	(Intercept)	-1.4017206	0.0069810	-200.79138	0	***
Entertainment/Recreation	age_group19_65	0.8625695	0.0083249	103.61305	0	***
Entertainment/Recreation	age_group65_plus	-0.3668170	0.0109116	-33.61707	0	***
Personal Services	(Intercept)	-1.4113648	0.0072879	-193.65900	0	***
Personal Services	age_group19_65	0.8613305	0.0086925	99.08905	0	***
Personal Services	age_group65_plus	-0.3075627	0.0111961	-27.47049	0	***
Hospitality/Lodging	(Intercept)	-1.3693572	0.0143591	-95.36538	0	***
Hospitality/Lodging	age_group19_65	0.8641156	0.0172122	50.20355	0	***
Hospitality/Lodging	age_group65_plus	-0.5964723	0.0239956	-24.85754	0	***
Office/Professional	(Intercept)	-1.4045851	0.0039776	-353.11968	0	***

category	term	estimate	std.error	statistic	p.value	significance
Office/Professional	age_group19_65	0.8331159	0.0047644	174.86345	0	***
Office/Professional	age_group65_plus	-0.2570142	0.0060233	-42.67006	0	***
Restaurant/Bar	(Intercept)	-1.3871536	0.0054690	-253.63913	0	***
Restaurant/Bar	age_group19_65	0.8254640	0.0065583	125.86495	0	***
Restaurant/Bar	age_group65_plus	-0.3278165	0.0084512	-38.78927	0	***