OGPe Construction Permits Dataset - Cleaning & Analysis

Jack Keller (PR Studio 2024, Group 3)

2024-04-01

Overview

This document includes the code used to clean the updated OGPe Construction Permits dataset along with summaries and plots of the cleaned data.

The updated dataset includes data through November 2023.

The cleaned data includes information on Public, Private, Public with Private Contracting, and Public-Private Alliance permit types.

Cleaning

Raw data contains 33,505 rows and 20 variables.

```
### Translate from Spanish to English

# Since the dataset is in Spanish, we need to translate some data to English.
# We'll start with the column names.

permits_clean <- permits_raw
colnames(permits_clean) <- c('case', 'permit_acronym', 'filed_date', 'approval_status', 'issued_date', 'cadastr
e', 'municipality', 'lon', 'lat', 'x', 'y', 'physical_address', 'name', 'description', 'owner', 'project_type',
'procedure', 'rural_urban', 'cost_estimate', 'filed_by_pa')

# translate rural_urban values to English
# check for distinct values first
permits_clean %>% distinct(rural_urban)
```

```
## # A tibble: 2 × 1
##   rural_urban
##   <chr>
## 1 Rural
## 2 Urbano
```

```
# replace with English
permits_clean <- permits_clean %>%
  mutate(rural_urban = str_replace(rural_urban, "Urbano", "Urban"))
# check that it worked
permits_clean %>% distinct(rural_urban)
```

```
## # A tibble: 2 × 1
##   rural_urban
##   <chr>
## 1 Rural
## 2 Urban
```

```
# translate project_type values to English
# check for distinct values first
permits_clean %>% distinct(project_type)
```

```
## # A tibble: 4 × 1
##   project_type
##   <chr>
## 1 Privada
## 2 Público con Contratación Privada
## 3 Público
## 4 Alianza Público-Privada
```

```
# replace with English
permits_clean <- permits_clean %>%
  mutate(project_type = str_replace(project_type, "Privada", "Private")) %>%
  mutate(project_type = str_replace(project_type, "Público", "Public")) %>%
  mutate(project_type = str_replace(project_type, "Public con Contratación Private", "Public with Private Contrac
ting")) %>%
  mutate(project_type = str_replace(project_type, "Alianza Public-Private", "Public-Private Alliance"))
# check that it worked
permits_clean %>% distinct(project_type)
```

```
## # A tibble: 4 × 1
##   project_type
##   <chr>
## 1 Private
## 2 Public with Private Contracting
## 3 Public
## 4 Public-Private Alliance
```

```
# translate filed_by_pa values to English
# check for distinct values first
permits_clean %>% distinct(filed_by_pa)
```

```
## # A tibble: 2 × 1
##   filed_by_pa
##   <chr>
## 1 SÍ
## 2 NO
```

```
# replace with English
permits_clean <- permits_clean %>%
  mutate(filed_by_pa = str_replace(filed_by_pa, "SÍ", "yes")) %>%
  mutate(filed_by_pa = str_replace(filed_by_pa, "NO", "no"))
# check that it worked
permits_clean %>% distinct(filed_by_pa)
```

```
## # A tibble: 2 × 1
##   filed_by_pa
##   <chr>
## 1 yes
## 2 no
```

```
# save the English version of the raw data
# convert date variables to character type so they export to csv correctly
permits_raw_eng <- permits_clean
permits_raw_eng$filed_date <- as.Date.character(permits_raw_eng$filed_date, format='%Y-%m-%d')
permits_raw_eng$issued_date <- as.Date.character(permits_raw_eng$filed_date, format='%Y-%m-%d')

write_excel_csv(permits_raw_eng, '../Datasets/OGPE/Raw/OGPE_ConstructionPermits_UpdatedThroughNov23_Raw_ENG.csv')
```

```
### Clean data
# remove duplicate rows
init_row_num <- nrow(permits_clean)
permits_clean <- permits_clean %>%
  distinct(.keep_all = TRUE)
final_row_num <- nrow(permits_clean)
init_row_num == final_row_num # TRUE, no duplicates removed
```

```
## [1] TRUE
```

```
# check for duplicates based on case
init_row_num <- nrow(permits_clean)
permits_clean <- permits_clean %>%
  distinct(case, .keep_all = TRUE)
final_row_num <- nrow(permits_clean)
init_row_num == final_row_num # TRUE, no duplicates removed
```

```
## [1] TRUE
```

```
# check for rows where case # is different but everything else is the same
init_row_num <- nrow(permits_clean)
permits_clean <- permits_clean[!(duplicated(permits_clean[-1]) | duplicated(permits_clean[-1], fromLast = TRU
E)),]
final_row_num <- nrow(permits_clean)

init_row_num == final_row_num # FALSE, duplicates removed
```

```
## [1] FALSE
```

```
init_row_num - final_row_num # 64 duplicates found
```

```
## [1] 64
```

```
# 64 duplicate rows removed
# Note: we don't check for duplicate location, because multiple projects/permits could have occurred at the same
address/building

###

# Check for missing values
# Let's deal with missing values first
summary(permits_clean) # see where missing values (NA) exist
```

```
##      case           permit_acronym       filed_date
##  Length:33441       Length:33441       Min.   :2014-12-02 00:00:00.00
##  Class :character   Class :character   1st Qu.:2018-11-20 00:00:00.00
##  Mode  :character   Mode  :character   Median :2020-12-31 00:00:00.00
##                                        Mean   :2020-06-05 20:00:50.38
##                                        3rd Qu.:2022-04-29 00:00:00.00
##                                        Max.   :2023-11-22 00:00:00.00
##
##  approval_status     issued_date                      cadastre
##  Length:33441       Min.   :2014-12-16 00:00:00.00   Length:33441
##  Class :character   1st Qu.:2019-03-16 00:00:00.00   Class :character
##  Mode  :character   Median :2021-04-21 00:00:00.00   Mode  :character
##                     Mean   :2020-09-21 05:28:48.78
##                     3rd Qu.:2022-08-12 00:00:00.00
##                     Max.   :2023-11-30 00:00:00.00
##
##  municipality            lon               lat              x
##  Length:33441       Min.   :-68.43   Min.   :17.78   Min.   :111712
##  Class :character   1st Qu.:-66.86   1st Qu.:18.15   1st Qu.:155306
##  Mode  :character   Median :-66.30   Median :18.34   Median :214388
##                     Mean   :-66.42   Mean   :18.28   Mean   :200983
##                     3rd Qu.:-66.07   3rd Qu.:18.42   3rd Qu.:238800
##                     Max.   :-63.26   Max.   :18.51   Max.   :324977
##                     NA's   :178      NA's   :178
##        y          physical_address       name           description
##  Min.   :211130   Length:33441       Length:33441       Length:33441
##  1st Qu.:235568   Class :character   Class :character   Class :character
##  Median :255739   Mode  :character   Mode  :character   Mode  :character
##  Mean   :249877
##  3rd Qu.:264782
##  Max.   :275512
##
##     owner           project_type        procedure         rural_urban
##  Length:33441       Length:33441       Length:33441       Length:33441
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
```

```
##
##  cost_estimate       filed_by_pa
##  Min.   :   -59600   Length:33441
##  1st Qu.:    21600   Class :character
##  Median :    58928   Mode  :character
##  Mean   :   267638
##  3rd Qu.:   155942
##  Max.   :179675453
##
```

```
# 178 NAs in lon and lat
# Check that missing lons/lats occur on the same rows
which(!which(is.na(permits_clean$lon))==which(is.na(permits_clean$lat)))
```

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

```
# Yes, they occur on the same rows (no FALSE entries)
# Because all entries have x/y spatial coordinates, we will map permits using these instead.

###

# Next let's look at impossible values for cost
# hist(permits_clean$cost_estimate)

filter(permits_clean, cost_estimate <= 0)
```

```
## # A tibble: 921 × 20
##    case   permit_acronym filed_date          approval_status issued_date
##    <chr>  <chr>          <dttm>              <chr>           <dttm>
##  1 2023–… PCOC           2023-11-20 00:00:00 PERMIT_APPROVED 2023-11-22 00:00:00
##  2 2023–… PCOC           2023-11-17 00:00:00 PERMIT_PRE_APP… 2023-11-30 00:00:00
##  3 2023–… PCOC           2023-11-10 00:00:00 PERMIT_APPROVED 2023-11-14 00:00:00
##  4 2022–… PCOC           2023-11-06 00:00:00 PERMIT_APPROVED 2023-11-10 00:00:00
##  5 2020–… PCOC           2023-10-19 00:00:00 PERMIT_APPROVED 2023-11-03 00:00:00
##  6 2023–… PCOC           2023-10-17 00:00:00 PERMIT_APPROVED 2023-10-27 00:00:00
##  7 2023–… PCOC           2023-10-17 00:00:00 PERMIT_APPROVED 2023-10-25 00:00:00
##  8 2021–… PCOC           2023-10-13 00:00:00 PERMIT_APPROVED 2023-10-16 00:00:00
##  9 2023–… PCOC           2023-10-12 00:00:00 PERMIT_APPROVED 2023-11-14 00:00:00
## 10 2022–… PCOC           2023-10-12 00:00:00 PERMIT_APPROVED 2023-10-26 00:00:00
## # ℹ 911 more rows
## # ℹ 15 more variables: cadastre <chr>, municipality <chr>, lon <dbl>,
## #   lat <dbl>, x <dbl>, y <dbl>, physical_address <chr>, name <chr>,
## #   description <chr>, owner <chr>, project_type <chr>, procedure <chr>,
## #   rural_urban <chr>, cost_estimate <dbl>, filed_by_pa <chr>
```

```
# 921 rows are missing cost (cost_estimate <= 0)
# we will replace all of these values with 0 to denote missing cost
permits_clean$cost_estimate <- replace(permits_clean$cost_estimate, permits_clean$cost_estimate < 0, 0)
filter(permits_clean, cost_estimate == 0) # check that it worked
```

```
## # A tibble: 921 × 20
##    case   permit_acronym filed_date          approval_status issued_date
##    <chr>  <chr>          <dttm>              <chr>           <dttm>
##  1 2023-… PCOC           2023-11-20 00:00:00 PERMIT_APPROVED 2023-11-22 00:00:00
##  2 2023-… PCOC           2023-11-17 00:00:00 PERMIT_PRE_APP… 2023-11-30 00:00:00
##  3 2023-… PCOC           2023-11-10 00:00:00 PERMIT_APPROVED 2023-11-14 00:00:00
##  4 2022-… PCOC           2023-11-06 00:00:00 PERMIT_APPROVED 2023-11-10 00:00:00
##  5 2020-… PCOC           2023-10-19 00:00:00 PERMIT_APPROVED 2023-11-03 00:00:00
##  6 2023-… PCOC           2023-10-17 00:00:00 PERMIT_APPROVED 2023-10-27 00:00:00
##  7 2023-… PCOC           2023-10-17 00:00:00 PERMIT_APPROVED 2023-10-25 00:00:00
##  8 2021-… PCOC           2023-10-13 00:00:00 PERMIT_APPROVED 2023-10-16 00:00:00
##  9 2023-… PCOC           2023-10-12 00:00:00 PERMIT_APPROVED 2023-11-14 00:00:00
## 10 2022-… PCOC           2023-10-12 00:00:00 PERMIT_APPROVED 2023-10-26 00:00:00
## # ℹ 911 more rows
## # ℹ 15 more variables: cadastre <chr>, municipality <chr>, lon <dbl>,
## #   lat <dbl>, x <dbl>, y <dbl>, physical_address <chr>, name <chr>,
## #   description <chr>, owner <chr>, project_type <chr>, procedure <chr>,
## #   rural_urban <chr>, cost_estimate <dbl>, filed_by_pa <chr>
```

```
###

# Next map locations to Census Block Group level (generate FIPS codes)

# Convert x/y Cartesian coordinates to sf format (sf library)
# spatial mapping code used: https://epsg.io/4437
# accurate to 2.0 m
spatial_coords <- st_as_sf(permits_clean, coords=c('x', 'y'), crs='EPSG:4437')[c('geometry')]

# Now convert x/y spatial coordinates to FIPS codes down to the Census Block level
# tutorial: https://shiandy.com/post/2020/11/02/mapping-lat-long-to-fips/
# census data source: https://catalog.data.gov/dataset/tiger-line-shapefile-current-state-puerto-rico-block-group
census_block_groups <- st_read('../Datasets/Census/Block Groups 2023/tl_2023_72_bg/tl_2023_72_bg.shp', quiet=TRU
E)

# census data uses spatial code EPSG:4269 - https://epsg.io/4269-1731
#st_crs(census_block_groups)

# transform coords to spatial code used by census data
spatial_coords <- st_transform(spatial_coords, crs=st_crs(census_block_groups))

intersected <- st_intersects(spatial_coords, census_block_groups)

sum(is.na(intersected))
```

```
## [1] 0
```

```r
# 0 NA rows: none were outside range of census block groups
# So, no need to exclude rows based on invalid coordinates

# get fips codes
spatial_coords <- spatial_coords %>%
  mutate(intersection = as.integer(intersected),
         fips = if_else(is.na(intersection), "",
                        census_block_groups$GEOID[intersection]))

# add fips variable to permits data
permits_clean <- permits_clean %>%
  add_column(cbg_fips = spatial_coords$fips, .after = 'y')

###

# add duration variable
# first convert date variables to Date format
filed <- as.Date(permits_clean$filed_date)
issued <- as.Date(permits_clean$issued_date)

# calculate durations = issued_date - filed_date (in days)
durations <- as.numeric(difftime(issued, filed, units='days'))

# add durations to permits data
permits_clean <- permits_clean %>%
  add_column(duration = durations, .after = 'issued_date')

nrow(permits_clean)
```

```
## [1] 33441
```

```r
# 33,441 rows remain
```

```
# add binary-encoded category variables
# this code is adapted from the 2023 studio group (we assumed the same categories to capture permit descriptions)
# see 'Datasets/OGPE Permit Data - 2023 Studio/PRStudio_2023_DataProcessing.r', lines 193-307 for original code

# we converted the code to use regular expressions (regex) which are more concise
# Note: matching is case-INSENSITIVE

# Hurricane Maria
permits_clean$maria <-  if_else(grepl("maria", permits_clean$description, ignore.case=TRUE), 1, 0)

# CDBG/R3 Program
permits_clean$cdbg_r3 <-  if_else(grepl("r3|cdgb|ver memorial explicativo", permits_clean$description, ignore.cas
e=TRUE), 1, 0)

# Construction, general
permits_clean$construction <-  if_else(grepl("permiso de construccion|construccion|construcion", permits_clean$de
scription,
                                                    ignore.case=TRUE), 1, 0)

# Residential, general
permits_clean$residential <-  if_else(grepl("residencia|vivienda|recidencia", permits_clean$description, ignore.c
ase=TRUE), 1, 0)

# Single Family Residential
permits_clean$sf_residential <-  if_else(grepl("residencia|vivienda|construccion de casa|unifamiliar|una planta|u
n nivel|una
                                                      familia", permits_clean$description, ignore.case=TRUE),
1, 0)

# Multifamily Residential
permits_clean$mf_residential <-  if_else(grepl("multifamiliar|apartamento|apartamentos|duplex|condominio",
                                                      permits_clean$description, ignore.case=TRUE), 1, 0)

# Remodel, general
permits_clean$remodel <-  if_else(grepl("remodelacion|sustitucion|sustituir|mejoras|remodelar", permits_clean$des
cription,
                                                ignore.case=TRUE), 1, 0)

# Residential Remodel
```

```
permits_clean$residential_remodel <-  if_else(grepl("remodelacion|residencia|vivienda", permits_clean$descriptio
n,
                                                                    ignore.case=TRUE), 1, 0)

# Residential Expansion
permits_clean$residential_expansion <-  if_else(grepl("ampliacion|expansion|residencia|vivienda", permits_clean$d
escription,
                                                                      ignore.case=TRUE), 1, 0)

# Expansion, general
permits_clean$expansion <-  if_else(grepl("ampliacion|expansion|accesorio|construccion de segunda|construccion de
tercer|adicion|2da|2do|segunda planta", permits_clean$description, ignore.case=TRUE), 1, 0)

# Demolition and Reconstruction
permits_clean$demo_reconstr <-  if_else(grepl("demolicion|remodelacion|reconstruccion", permits_clean$descriptio
n, ignore.case=TRUE), 1, 0)

# Demolition
permits_clean$demolition <-  if_else(grepl("demolicion|demolition", permits_clean$description, ignore.case=TRUE),
1, 0)

# Repairs
permits_clean$repair <-  if_else(grepl("reemplazo|rehabilitacion|reparacion|rehabilitar", permits_clean$descripti
on, ignore.case=TRUE), 1, 0)

# Housing Development
permits_clean$housing_develop <-  if_else(grepl("proyecto residencial|urbanizacion|desarrollo residencial|solares
residencial|urb.",
                                          permits_clean$description, ignore.case=TRUE), 1, 0)
# Trailer
permits_clean$trailer <-  if_else(grepl("trailer|vagon|camper", permits_clean$description, ignore.case=TRUE), 1,
0)

# Commercial
permits_clean$commercial <-  if_else(grepl("comercial|comercio|restaurant|food
truck|turistico|tienda|gasolina|negocio|venta|dispensario|almacen|pantalla
digital|anuncio|industrial|llc|manufactura|asfalto|joyeria|hotel|billboard|veterinaria|burger|coffee|cafeteria|co
mpania|lavanderia|retail|farmacia", permits_clean$description, ignore.case=TRUE), 1, 0)
```

```
# Pool
permits_clean$pool <-  if_else(grepl("piscina", permits_clean$description, ignore.case=TRUE), 1, 0)

# Utilities
permits_clean$utils <-  if_else(grepl("telecomunicaciones|utilities|at&t|septico|fibra optica|poste de hormigon",
                                      permits_clean$description, ignore.case=TRUE), 1, 0)

# Legalization
permits_clean$legal <-  if_else(grepl("legalizar|legalizacion|legalizacion de", permits_clean$description, ignor
e.case=TRUE), 1, 0)

# Residential Legalization
permits_clean$residential_legal <-  if_else(grepl("legalizar residencia|legalizacion|legalizacion de|residencia",
                                                  permits_clean$description, ignore.case=TRUE), 1, 0)

# Community Space
permits_clean$community <-  if_else(grepl("cancha|baloncesto|recreativa|iglesia|biblioteca|parque|centro
                                                     comunal|estadio|atletica|park", permits_clean$description, i
gnore.case=TRUE), 1, 0)

# Government
permits_clean$govt <-  if_else(grepl("alcadia|gobierno municipal|municipio", permits_clean$description, ignore.ca
se=TRUE), 1, 0)

# Public Services
permits_clean$public_serv <-  if_else(grepl("medico|hospital|medica|dental", permits_clean$description, ignore.ca
se=TRUE), 1, 0)

# Solar Power
permits_clean$solar <-  if_else(grepl("fotovoltaico|placas solares|sistema solar", permits_clean$description, ign
ore.case=TRUE), 1, 0)

# Minor Construction
permits_clean$minor_constr <-  if_else(grepl("terraza|verja|marquesina", permits_clean$description, ignore.case=T
RUE), 1, 0)

# Parcel Split
permits_clean$parcel_split <- if_else(grepl("segregacion|lotificacion|dividir|subdivision", permits_clean$descrip
tion, ignore.case=TRUE), 1, 0)
```

```
# put rows which were not selected for any above category in 'other' category
permits_clean$other <- as.numeric(rowSums(permits_clean[, (ncol(permits_clean)-25):ncol(permits_clean)]) == 0)
sum(permits_clean$other == 1)
```

```
## [1] 4556
```

```
# 3951 rows (13.05%) marked 'other'
# TODO: This seems a little high (compared to 2023 studio's ~4% marked 'other'), maybe look into it.

###

# create binary-encoded corporate_owner variable
# predict whether permit is Corporate/Business owned by using keyword identifiers (using Regex)
new_col <- if_else(
  grepl('(llc|l\\.l\\.c\\.|ltd|incorporated|inc\\.|corp\\.|enterprise|investment|company|compania|dbr|dba|group|c
o\\.)',
        permits_clean$owner,
          ignore.case = TRUE),
            1,0)
sum(new_col == 1) # 4750 permits flagged as corporate/business owned (15.69%)
```

```
## [1] 4781
```

```
# add new variable to permits data
permits_clean <- permits_clean %>%
  add_column(corporate_owner = new_col, .after = 'owner')

###

# convert date variables to character type so they export to csv correctly
permits_clean$filed_date <- as.Date.character(permits_clean$filed_date, format='%Y-%m-%d')
permits_clean$issued_date <- as.Date.character(permits_clean$filed_date, format='%Y-%m-%d')

# save clean data as csv
write_excel_csv(permits_clean, '../Datasets/OGPE/Clean/OGPE_ConstructionPermits_UpdatedThroughNov23_CLEAN.csv')
```

Analysis

Cleaned dataset includes **33,441 rows** and **50 variables.**

Data cleaning process included:
- Translated the variable names and select columns to English from Spanish.
- Noted 178 rows missing latitude/longitude coordinates (relied on x/y spatial coordinates instead).
- Generated FIPS codes for each permit based on x/y coordinates down to the census block group level.
- Replaced 921 missing costs (2.75%) with the value 0.
- Removed 64 duplicates (case # differed but all other variables were the same).
- Created duration variable (in days).
- Created 27 categories variables based on description (including 'other' category).
- Created corporate_owner variable which predicts whether the owner of a permit is a corporation or business (15.69% of permits).

(Created 30 new variables total).

GIS Prep

Exporting CSVs with spatial information related to permits filed in individual reconstruction periods following each disaster.

```
# write all private and private-related permit coordinates to csv file
# these will be used for mapping
# NOTE:
# spatial mapping code used: https://epsg.io/4437
# accurate to 2.0 m

unique(permits_clean$approval_status)
```

```
## [1] "PERMIT_APPROVED"     "PERMIT_PRE_APPROVED"
```

```
# all permits are either approved or pre-approved. So, no need to filter for approved only

# all private and private-related, approved permits since Hurricane Irma
private_coords <- permits_clean %>%
  filter(.$project_type != "Public") %>%
  filter(.$filed_date > '2017-09-06') %>% # filed after Hurricane Irma hit on Sept 6, 2017
  select(c('x', 'y'))

write_excel_csv(private_coords, '../Datasets/OGPE/Clean/Spatial Data/private_related_coordinates_post_irma_EPSG_4
437.csv')

### INDIVIDUAL RECONSTRUCTION PERMITS
# private and private-related, approved permits 1 year after Irma/Maria
private_coords <- permits_clean %>%
  filter(.$project_type != "Public") %>%
  filter(.$filed_date > '2017-09-06' & .$filed_date <= '2018-09-06') %>% # filed within 1 year Hurricane Irma hit
on Sept 6, 2017 (includes Maria)
  select(c('x', 'y'))

write_excel_csv(private_coords, '../Datasets/OGPE/Clean/Spatial Data/private_related_coordinates_1yr_post_irma_ma
ria_EPSG_4437.csv')

# private and private-related, approved permits 1 year after 2020 Earthquakes
private_coords <- permits_clean %>%
  filter(.$project_type != "Public") %>%
  filter(.$filed_date > '2020-01-07' & .$filed_date <= '2021-01-07') %>% # filed within 1 year after main earthqu
ake hit on Jan 7, 2020
  select(c('x', 'y'))

write_excel_csv(private_coords, '../Datasets/OGPE/Clean/Spatial Data/private_related_coordinates_1yr_post_earthqu
akes_EPSG_4437.csv')

# private and private-related, approved permits 1 year after Hurricane Fiona
private_coords <- permits_clean %>%
  filter(.$project_type != "Public") %>%
  filter(.$filed_date > '2022-09-18' & .$filed_date <= '2023-09-18') %>% # filed within 1 year after Hurricane Fi
ona hit on Sept 18, 2022
  select(c('x', 'y'))
```

```
write_excel_csv(private_coords, '../Datasets/OGPE/Clean/Spatial Data/private_related_coordinates_1yr_post_fiona_EPSG_4437.csv')
```

Site Visit Query

```
sj_permits <- permits_clean %>%
  filter(.$municipality == "San Juan") %>%
  filter(.$approval_status == "PERMIT_APPROVED") %>%
  filter(.$issued_date >= "2023-09-01") # filter for the last 6 months (want a project currently under construction)

nrow(sj_permits)
# 47 permits filed in San Juan, approved, and issued within the last 6 months

View(sj_permits)
```

Summary

```r
# generate table of descriptive statistics summarizing data
# using arsenal library

# narrow down relevant variables
permits <- permits_clean %>%
  select(duration, cost_estimate, project_type, rural_urban, filed_by_pa, corporate_owner)

# convert binary corporate_owner to yes/no
permits$corporate_owner <- factor(ifelse(permits$corporate_owner, "Yes", "No"))

# make filed_by_pa column uppercase for formatting purposes
permits$filed_by_pa <- factor(permits$filed_by_pa, levels=c("no", "yes"), labels=c("No", "Yes"))

# filter out missing costs, negative durations
init = nrow(permits)
permits <- permits %>%
  filter(.$cost_estimate > 0) %>%
  filter(.$duration >= 0)

final = nrow(permits)
init - final
```

```
## [1] 931
```

```
# 931 rows removed due to missing cost/negative duration

# summarize
permits_summary <- tableby( ~ ., data = permits)

labels <- list(
  duration = "Approval Time (days)",
  cost_estimate = "Estimated Cost ($)",
  project_type = "Project Type",
  rural_urban = "Rural/Urban",
  filed_by_pa = "Filed by PA",
  corporate_owner = "Corporate Owner"
)

summary(permits_summary, title = "Construction Permits Summary", labelTranslations = labels)
```

```
##
##
## Table: Construction Permits Summary
##
## |                                                |    Overall (N=32510)     |
## |:-----------------------------------------------|:------------------------:|
## |**Approval Time (days)**                        |                          |
## |   Mean (SD)                     |    108.951 (162.307)     |
## |   Range                         |     0.000 - 2480.000     |
## |**Estimated Cost ($)**                          |                          |
## |   Mean (SD)                     | 275299.203 (1689750.323) |
## |   Range                         |  0.210 - 179675453.000   |
## |**Project Type**                                |                          |
## |   Private                       |      29423 (90.5%)       |
## |   Public                        |        879 (2.7%)        |
## |   Public with Private Contracting |       2155 (6.6%)        |
## |   Public-Private Alliance       |        53 (0.2%)         |
## |**Rural/Urban**                                 |                          |
## |   Rural                         |      14494 (44.6%)       |
## |   Urban                         |      18016 (55.4%)       |
## |**Filed by PA**                                 |                          |
## |   No                            |      26434 (81.3%)       |
## |   Yes                           |       6076 (18.7%)       |
## |**Corporate Owner**                             |                          |
## |   No                            |      28004 (86.1%)       |
## |   Yes                           |       4506 (13.9%)       |
```

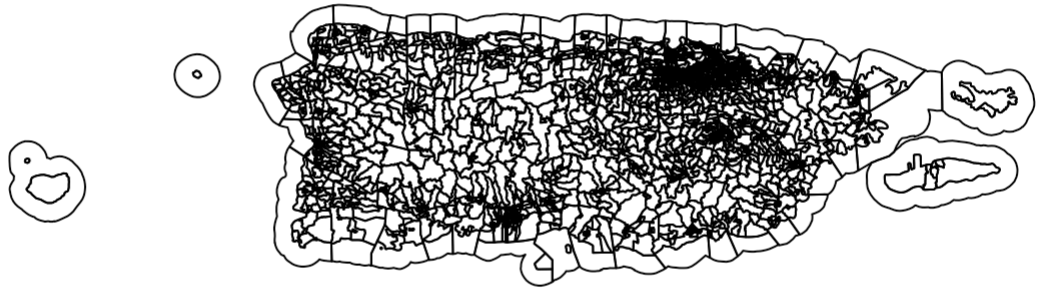
```
# CODE FOR PRIVATE ONLY ANALYSIS (Not used in final report)
# # remove public permits
# # (Private, Public w/ Private Contracting, Public-Private ALliance types remain)
# private_permits <- permits_clean %>%
#   filter(!permits_clean$project_type == "Public")
#
# # ? private permits
# nrow(private_permits)
#
# # Urban/Rural
# table(private_permits$rural_urban)
#
# # Rural
# # mean duration
# private_permits %>%
#   filter(private_permits$rural_urban == 'Rural') %>%
#   summarise(., mean(duration))
#
# # mean cost
# private_permits %>%
#   filter(private_permits$rural_urban == 'Rural') %>%
#   filter(.$cost_estimate > 0) %>%
#   summarise(., mean(cost_estimate))
#
# # Urban
# # mean duration
# private_permits %>%
#   filter(private_permits$rural_urban == 'Urban') %>%
#   summarise(., mean(duration))
#
# # mean cost
# private_permits %>%
#   filter(private_permits$rural_urban == 'Urban') %>%
#   filter(.$cost_estimate > 0) %>%
#   summarise(., mean(cost_estimate))
#
#
# # Filed by PA
# table(private_permits$filed_by_pa)
```

```
#
# # yes
# # mean duration
# private_permits %>%
#   filter(private_permits$filed_by_pa == 'yes') %>%
#   summarise(., mean(duration))
#
# # mean cost
# private_permits %>%
#   filter(private_permits$filed_by_pa == 'yes') %>%
#   filter(.$cost_estimate > 0) %>%
#   summarise(., mean(cost_estimate))
#
# # no
# # mean duration
# private_permits %>%
#   filter(private_permits$filed_by_pa == 'no') %>%
#   summarise(., mean(duration))
#
# # mean cost
# private_permits %>%
#   filter(private_permits$filed_by_pa == 'no') %>%
#   filter(.$cost_estimate > 0) %>%
#   summarise(., mean(cost_estimate))
#
#
# # Corporate owner
# table(private_permits$corporate_owner)
#
# # yes (1)
# # mean duration
# private_permits %>%
#   filter(private_permits$corporate_owner == 1) %>%
#   summarise(., mean(duration))
#
# # mean cost
# private_permits %>%
#   filter(private_permits$corporate_owner == 1) %>%
#   filter(.$cost_estimate > 0) %>%
```

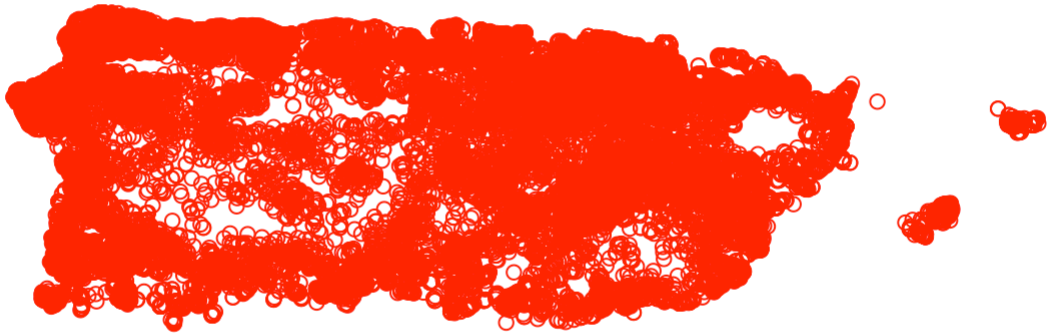
```
#   summarise(., mean(cost_estimate))
#
# # no (0)
# # mean duration
# private_permits %>%
#   filter(private_permits$corporate_owner == 0) %>%
#   summarise(., mean(duration))
#
# # mean cost
# private_permits %>%
#   filter(private_permits$corporate_owner == 0) %>%
#   filter(.$cost_estimate > 0) %>%
#   summarise(., mean(cost_estimate))
```

Maps

```
census_block_groups %>%
  select(geometry) %>%
    plot()
```



```
spatial_coords %>%
  select(geometry) %>%
      plot(reset=FALSE, col = "red")
```



Plots

```
# features include cost, duration
# also want to keep rural_urban, project_type, and filed_by_pa variables for comparison

total_count = nrow(permits)
total_removed = 0

# grab necessary variables
permits <- permits_clean %>%
  select(duration, cost_estimate, project_type, rural_urban, filed_by_pa, corporate_owner)

# convert binary corporate_owner to yes/no
permits$corporate_owner <- factor(ifelse(permits$corporate_owner, "Yes", "No"))

# make filed_by_pa column uppercase for formatting purposes
permits$filed_by_pa <- factor(permits$filed_by_pa, levels=c("no", "yes"), labels=c("No", "Yes"))

# filter out missing costs, negative durations
init = nrow(permits)
permits <- permits %>%
  filter(.$cost_estimate > 0) %>%
  filter(.$duration >= 0)

final = nrow(permits)
removed = init - final

total_removed = total_removed + removed
total_removed
```

```
## [1] 931
```

```
# 931 rows removed due to missing cost/negative duration

# remove outliers (> $50 million cost, > 2000 duration)
init = nrow(permits)
permits <- permits %>%
  filter(.$cost_estimate <= 50000000) %>%
  filter(.$duration <= 2000)

final = nrow(permits)
removed = init - final
removed
```

```
## [1] 13
```

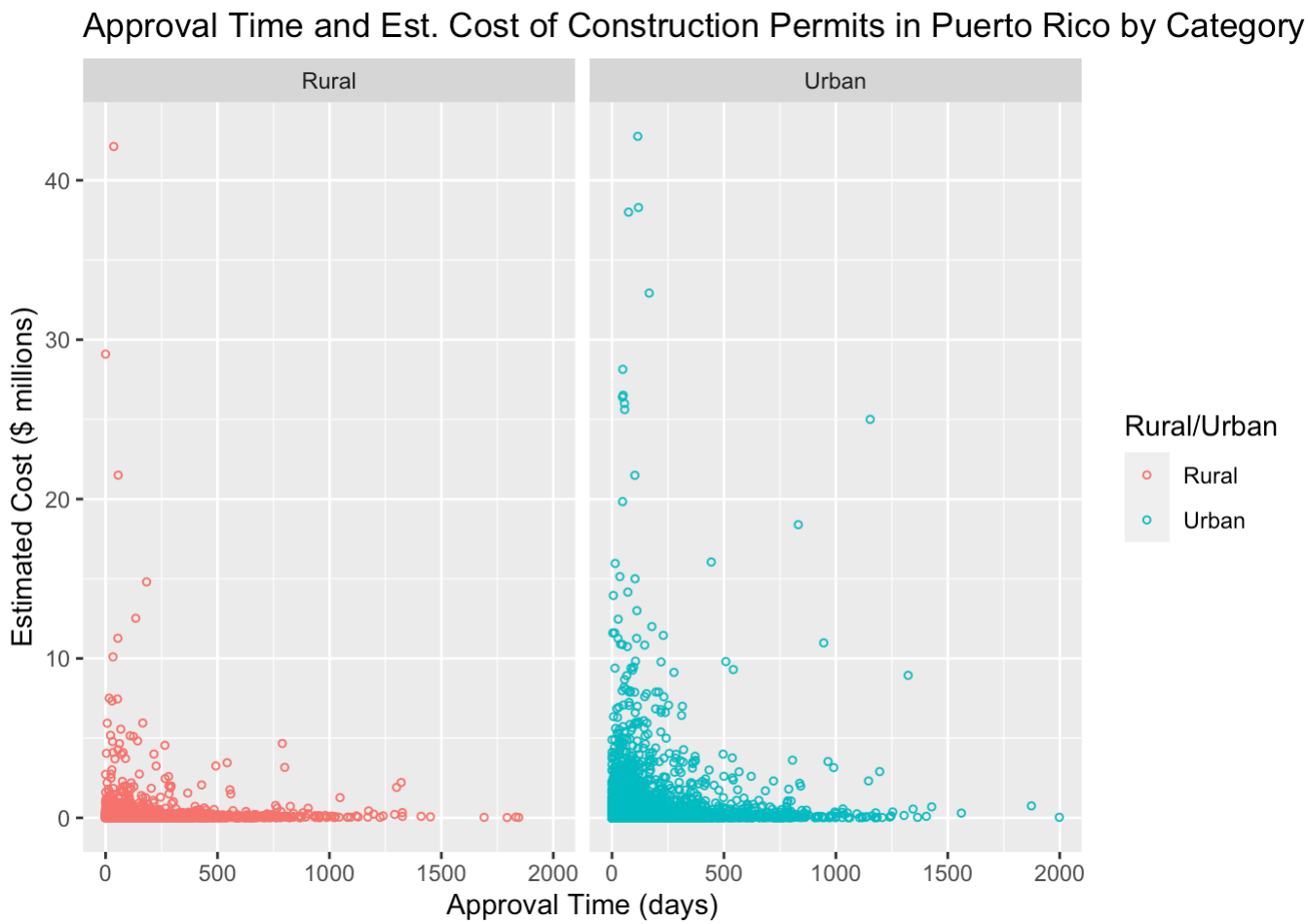
```
total_removed = total_removed + removed
# 13 outliers removed

total_removed / total_count
```

```
## [1] 0.02903722
```

```
# 5.81% of the dataset removed

# sample 50% of the remaining rows (without replacement)
permits <- sample_frac(permits, 0.50)
nrow(permits)
```

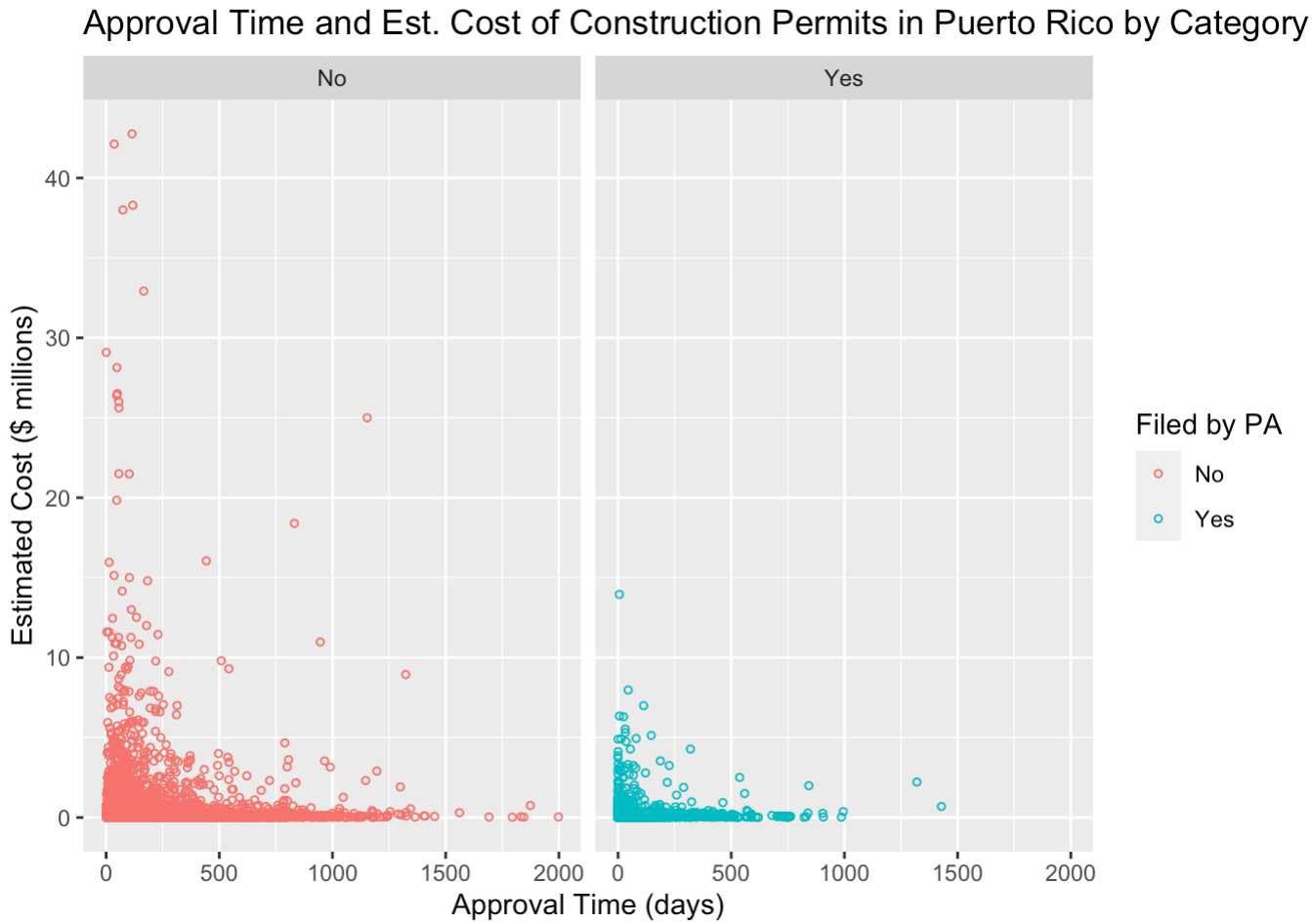
```
## [1] 16248
```

```r
# 16248 rows plotted

# 50% of 94.19% -> 47.095% of rows plotted -> ~47.1%

# creates scatter plots based on 'category' variable
create_scatter_plot <- function(category, facet_var, legend_title) {
  ggplot(permits) +
    scale_y_continuous(labels = c(0, 10, 20, 30, 40)) +
    geom_jitter(aes(x=duration, y=cost_estimate, color=category), shape=1, size=1) +
    facet_wrap(facet_var) +
    labs(title="Approval Time and Est. Cost of Construction Permits in Puerto Rico by Category", x="Approval Time
(days)", y="Estimated Cost ($ millions)", color=legend_title)
}

# create scatter plots
# rural/urban:
create_scatter_plot(permits$rural_urban, ~permits$rural_urban, "Rural/Urban")
```

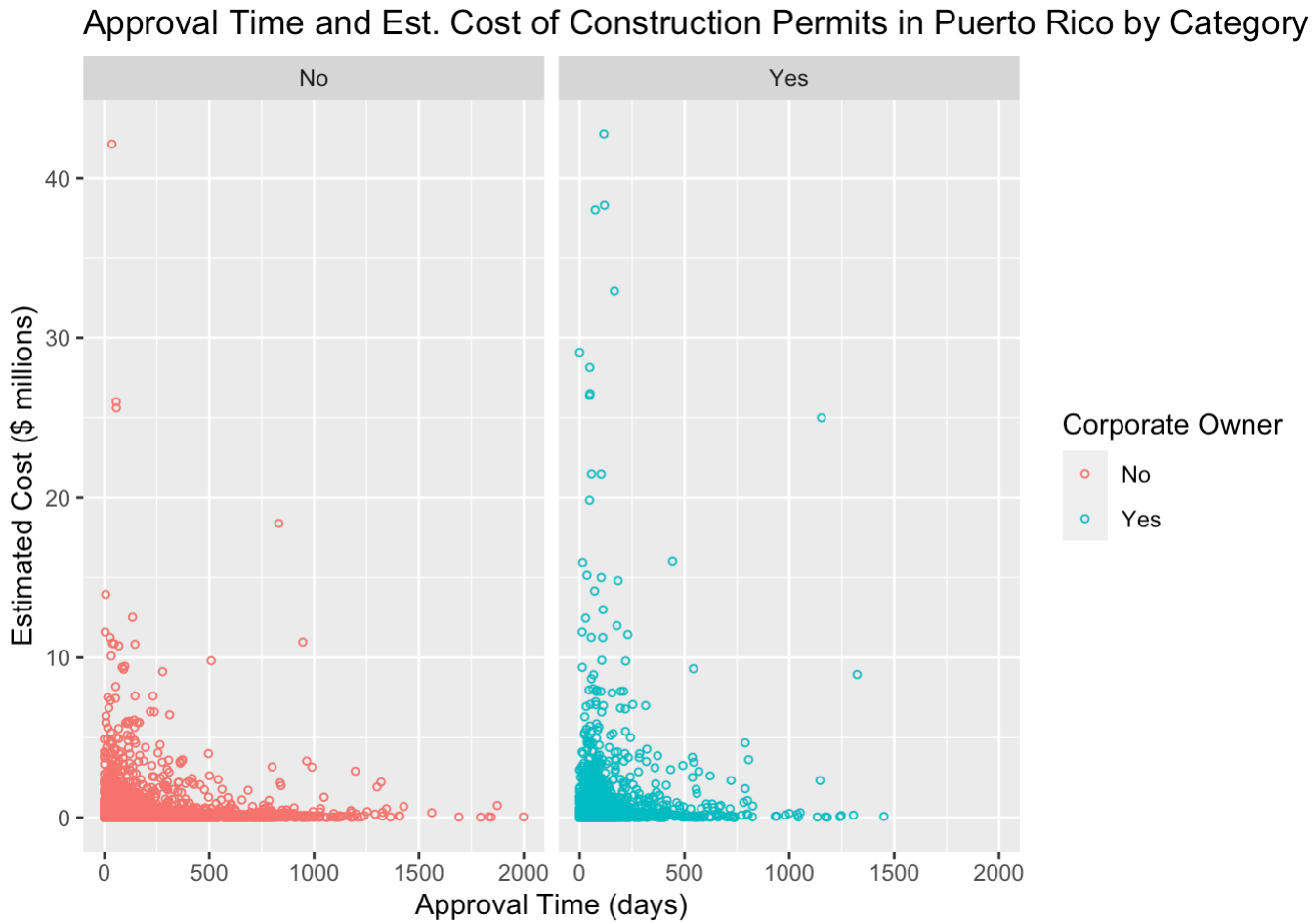



```
# project type:
create_scatter_plot(permits$project_type, ~permits$project_type, "Project Type")
```


```
# filed by pa:
create_scatter_plot(permits$filed_by_pa, ~permits$filed_by_pa, "Filed by PA")
```

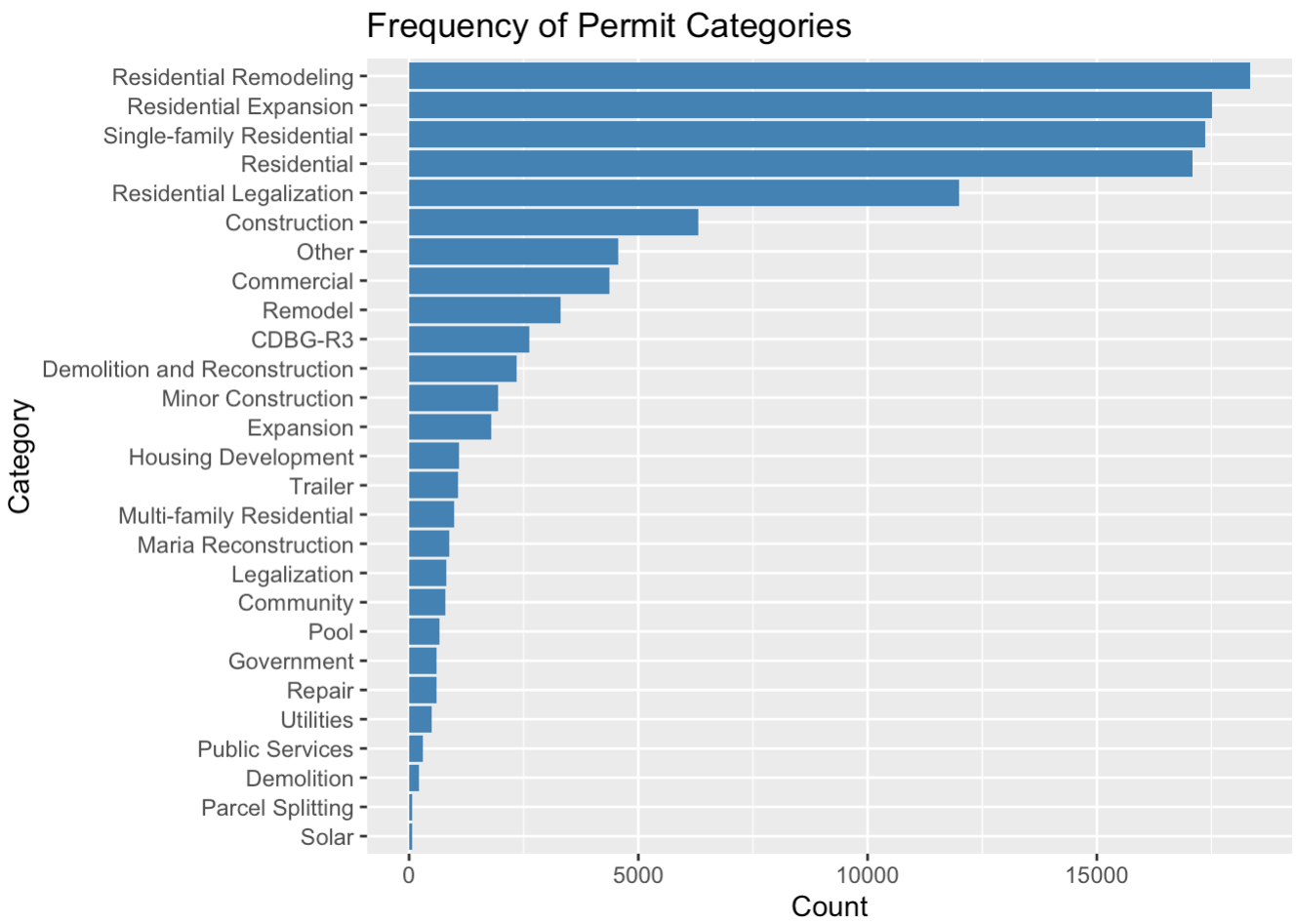



```
# corporate owner:
create_scatter_plot(permits$corporate_owner, ~permits$corporate_owner, "Corporate Owner")
```


Approval Time and Est. Cost of Construction Permits in Puerto Rico by Category
No
Yes
Estimated Cost ($ millions)
0
10
20
30
40
0
500
1000
1500
2000
Approval Time (days)
Corporate Owner
No
Yes

Charts

```r
# bar plot of permit categories (frequency)

# Get permit counts for categories
# Gather columns into long format
temp <- permits_clean[, 24:50] %>%
  pivot_longer(cols = maria:other, names_to = "column", values_to = "value")

# Filter for value == 1
temp <- temp %>%
  filter(value == 1)

# Get category counts
counts <- temp %>%
  group_by(column) %>%
  summarise(count = n())

# category labels in ascending order
cat_labels = c("Solar", "Parcel Splitting", "Demolition", "Public Services", "Utilities", "Repair", "Government",
"Pool", "Community", "Legalization", "Maria Reconstruction", "Multi-family Residential", "Trailer", "Housing Deve
lopment", "Expansion", "Minor Construction", "Demolition and Reconstruction", "CDBG-R3", "Remodel", "Commercial",
"Other", "Construction", "Residential Legalization", "Residential", "Single-family Residential", "Residential Exp
ansion", "Residential Remodeling")

# Plot counts in ascending order (largest on top)
ggplot(counts, aes(x=reorder(column, count), y=count)) +
  geom_bar(stat="identity", fill="steelblue") +
  scale_x_discrete(labels=cat_labels) +
  labs(title='Frequency of Permit Categories', x='Category', y='Count') +
  coord_flip()
```


Frequency of Permit Categories
Residential Remodeling
Residential Expansion
Single-family Residential
Residential
Residential Legalization
Construction
Other
Commercial
Remodel
CDBG-R3
Demolition and Reconstruction
Minor Construction
Expansion
Housing Development
Trailer
Multi-family Residential
Maria Reconstruction
Legalization
Community
Pool
Government
Repair
Utilities
Public Services
Demolition
Parcel Splitting
Solar
Category
0
5000
10000
15000
Count