

EDS 223 - HW 3: Identifying Impacts of Extreme Weather

Garrett Craig

2025-11-11

Table of contents

| | |
|---|----|
| INSERT README SCREENSHOT HERE | 1 |
| Overview | 1 |
| Data Sources | 2 |
| Setup | 2 |
| Load and Process Night Lights Data | 2 |
| Read and Combine VIIRS Tiles | 2 |
| Create Blackout Mask | 4 |
| Map: Night Lights Comparison | 6 |
| Exclude Highway Areas | 9 |
| Identify Affected Residential Buildings | 11 |
| Map: Houston Homes That Lost Power | 12 |
| Analyze Socioeconomic Impacts | 14 |
| Join with Census Data | 14 |
| Map: Blackout Events by Census Tract | 17 |
| Income Distribution Analysis | 18 |
| Results Summary | 20 |
| Discussion | 21 |
| Key Findings | 22 |
| Limitations | 22 |

INSERT README SCREENSHOT HERE

Overview

In February 2021, the state of Texas experienced unprecedented winter storms that caused widespread power outages across the state. This analysis uses remotely-sensed night lights data

to estimate the number of homes in Houston that lost power during these storms. By combining satellite data from NASA's Visible Infrared Imaging Radiometer Suite (VIIRS), building footprint data from OpenStreetMap, and socioeconomic data from the US Census Bureau, we can identify which areas were affected and explore potential disparities in infrastructure impacts.

Data Sources

- **VIIRS Night Lights Data:** Daily satellite imagery showing nighttime light intensity (4 tiles covering Houston area)
 - Pre-storm: February 7, 2021
 - During storm: February 16, 2021
- **OpenStreetMap (OSM):** Geospatial data on buildings and roads in Houston
- **US Census Bureau:** American Community Survey (ACS) 2019 5-year estimates at the census tract level

Setup

```
# Load required libraries
library(tidyverse)
library(sf)
library(stars)
library(tmap)
library(patchwork)

# Set options
options(scipen = 999)
```

Load and Process Night Lights Data

Read and Combine VIIRS Tiles

```
# Load night lights data for February 7, 2021 (pre-storm)
# First unzip the data if not already unzipped
if (!dir.exists("data/VNP46A1")) {
  unzip("data/VNP46A1.zip", exdir = "data")
}
```

```

# List all the tiles for Feb 7 (day 038) and Feb 16 (day 047)
# Note: files are in nested VNP46A1/VNP46A1/ directory
tiles_feb07 <- list.files("data/VNP46A1/VNP46A1",
                           pattern = "VNP46A1\\\.A2021038\\.h08v0[56]\\\.001.*\\.tif$",
                           full.names = TRUE)

tiles_feb16 <- list.files("data/VNP46A1/VNP46A1",
                           pattern = "VNP46A1\\\.A2021047\\.h08v0[56]\\\.001.*\\.tif$",
                           full.names = TRUE)

# Check that tiles were found
if (length(tiles_feb07) == 0) {
  stop("Oops! No VIIRS tiles found for February 7, 2021 (day 038). Please verify that:\n",
       "  - The VNP46A1 data is unzipped in the 'data/VNP46A1/VNP46A1/' directory\n",
       "  - Files match the pattern 'VNP46A1.A2021038.h08v0[56].001.*.tif'")
}

if (length(tiles_feb16) == 0) {
  stop("Oops! No VIIRS tiles found for February 16, 2021 (day 047). Please verify that:\n",
       "  - The VNP46A1 data is unzipped in the 'data/VNP46A1/VNP46A1/' directory\n",
       "  - Files match the pattern 'VNP46A1.A2021047.h08v0[56].001.*.tif'")
}

# Warn if unexpected number of tiles
if (length(tiles_feb07) != 2) {
  warning("Heads up! Expected 2 tiles for Feb 7, but found ", length(tiles_feb07),
          ". This may indicate missing or duplicate data files.")
}

if (length(tiles_feb16) != 2) {
  warning("Heads up! Expected 2 tiles for Feb 16, but found ", length(tiles_feb16),
          ". This may indicate missing or duplicate data files.")
}

# Read tiles as stars objects
feb07_tiles <- lapply(tiles_feb07, read_stars)
feb16_tiles <- lapply(tiles_feb16, read_stars)

# Check that all tiles have the same CRS
feb07_crs <- unique(sapply(feb07_tiles, function(x) st_crs(x)$wkt))
feb16_crs <- unique(sapply(feb16_tiles, function(x) st_crs(x)$wkt))

```

```

if (length(feb07_crs) > 1) {
  stop("Projection mismatch! The Feb 7 tiles have different coordinate reference systems.\n"
       "All tiles must be in the same projection to create a mosaic. Please reproject the data")
}

if (length(feb16_crs) > 1) {
  stop("Projection mismatch! The Feb 16 tiles have different coordinate reference systems.\n"
       "All tiles must be in the same projection to create a mosaic. Please reproject the data")
}

# Combine tiles for each date using st_mosaic
feb07_lights <- do.call(st_mosaic, feb07_tiles)
feb16_lights <- do.call(st_mosaic, feb16_tiles)

# Verify both mosaics have the same CRS
if (st_crs(feb07_lights) != st_crs(feb16_lights)) {
  stop("Cannot compare pre-storm and during-storm data! They have different projections:\n",
       " - Feb 7: ", st_crs(feb07_lights)$input, "\n",
       " - Feb 16: ", st_crs(feb16_lights)$input, "\n",
       "Both datasets must be in the same CRS to calculate light differences.")
}

```

Create Blackout Mask

```

# Calculate difference in light intensity between the two dates
light_diff <- feb07_lights - feb16_lights

# Check for any issues with the difference calculation
if (all(is.na(light_diff[[1]]))) {
  stop("Uh oh! The light difference calculation produced all NA values.\n",
       "This suggests the input rasters don't overlap or have incompatible extent/resolution\n",
       "Please check that feb07_lights and feb16_lights cover the same area.")
}

# Reclassify: identify areas with light drops > 200 nW cm^-2 sr^-1
# Areas with drops > 200 are considered blackouts (value = 1)
# All other areas get NA
blackout_mask <- light_diff > 200
blackout_mask[blackout_mask == FALSE] <- NA

```

```

# Check if any blackout areas were identified
n_blackout_pixels <- sum(!is.na(blackout_mask[[1]])), na.rm = TRUE)
if (n_blackout_pixels == 0) {
  warning("Hmm, no blackout areas detected with the 200 nW cm^-2 sr^-1 threshold.\n",
          " This seems unusual for the Feb 2021 Texas storms. Consider:\n",
          " - Lowering the threshold (e.g., try 150 or 100)\n",
          " - Checking that you're using the correct dates")
} else {
  cat(" Identified", n_blackout_pixels, "pixels with light drops > 200 nW cm^-2 sr^-1\n")
}

Identified 24846 pixels with light drops > 200 nW cm^-2 sr^-1

# Vectorize the raster
blackout_vector <- st_as_sf(blackout_mask)

# Check for invalid geometries before fixing
n_invalid <- sum(!st_is_valid(blackout_vector))
if (n_invalid > 0) {
  warning("Found ", n_invalid, " invalid geometries during vectorization.\n",
          " Don't worry - these will be automatically repaired using st_make_valid().")
}

# Fix invalid geometries
blackout_vector <- st_make_valid(blackout_vector)

# Define Houston bounding box coordinates
houston_coords <- matrix(c(-96.5, 29,
                            -96.5, 30.5,
                            -94.5, 30.5,
                            -94.5, 29,
                            -96.5, 29),
                           ncol = 2, byrow = TRUE)

# Create polygon and convert to sf object
# First create polygon, then convert to sfc, then to sf with CRS
houston_polygon <- st_polygon(list(houston_coords))
houston_sf <- st_sfc(houston_polygon, crs = st_crs(blackout_vector))

# Verify CRS match before spatial operation
if (st_crs(blackout_vector) != st_crs(houston_sf)) {
  stop("Cannot crop to Houston! The blackout data and Houston boundary have different project")
}

```

```

    " - Blackout: ", st_crs(blackout_vector)$input, "\n",
    " - Houston: ", st_crs(houston_sf)$input, "\n",
    "Use st_transform() to reproject one to match the other.")
}

# Crop blackout mask to Houston area
blackout_houston <- st_intersection(blackout_vector, houston_sf)

# Check if any blackout areas remain after cropping
if (nrow(blackout_houston) == 0) {
  warning("Unexpected result: No blackout areas found within the Houston bounding box!\n",
          "Please verify that:\n",
          " - Houston coordinates (-96.5 to -94.5°W, 29 to 30.5°N) are correct\n",
          " - The blackout data covers the Houston area\n",
          " - The CRS is correctly set (should be WGS84/EPSG:4326 for lat/lon)")
}

# Reproject to EPSG:3083 (NAD83 / Texas Centric Albers Equal Area)
blackout_houston <- st_transform(blackout_houston, crs = 3083)

```

Map: Night Lights Comparison

```

# Create side-by-side maps showing night lights before and after the storm
# Crop to Houston area for visualization
feb07_houston <- st_crop(feb07_lights, houston_sf)
feb16_houston <- st_crop(feb16_lights, houston_sf)
light_diff_houston <- st_crop(light_diff, houston_sf)

# Create Texas state boundary and Houston location for inset map
texas <- st_as_sf(maps::map("state", regions = "texas", plot = FALSE, fill = TRUE))
texas <- st_transform(texas, crs = 4326)
# Fix invalid geometries
texas <- st_make_valid(texas)
houston_bbox_4326 <- st_transform(houston_sf, crs = 4326)

# Create maps
tmap_mode("plot")

# Calculate shared breaks for consistent color scale
all_values <- c(as.vector(feb07_houston[[1]]), as.vector(feb16_houston[[1]]))

```

```

max_value <- quantile(all_values, 0.99, na.rm = TRUE)
breaks <- seq(0, max_value, length.out = 7)

# Create inset map of Texas showing Houston location
inset_map <- tm_shape(texas) +
  tm_polygons(col = "gray70", border.col = "gray30") +
  tm_shape(houston_bbox_4326) +
  tm_borders(col = "red", lwd = 3) +
  tm_layout(frame = TRUE,
            bg.color = "white",
            frame.lwd = 1,
            main.title = "Texas",
            main.title.size = 0.8,
            main.title.position = "center")

# Pre-storm map
map_feb07 <- tm_shape(feb07_houston) +
  tm_raster(title = "Light Intensity\n(nW cm-2 sr-1)",
             style = "fixed",
             breaks = breaks,
             palette = "YlOrRd",
             midpoint = NA) +
  tm_layout(main.title = "Pre-Storm (Feb 7)",
            main.title.size = 1.4,
            main.title.position = "center",
            main.title.fontface = "bold",
            legend.outside = FALSE,
            legend.position = c("right", "bottom"),
            legend.bg.color = "white",
            legend.bg.alpha = 0.8,
            legend.frame = TRUE,
            bg.color = "#2b2b2b",
            frame = FALSE) +
  tm_scale_bar(position = c("left", "bottom"),
               text.color = "black",
               bg.color = "white",
               bg.alpha = 0.7) +
  tm_compass(position = c("left", "top"),
             type = "4star",
             size = 2,
             color.dark = "black",
             color.light = "white",
             bg.color = "white",
             bg.alpha = 0.7)

```

```

    bg.color = "white",
    bg.alpha = 0.7)

# During storm map
map_feb16 <- tm_shape(feb16_houston) +
  tm_raster(title = "Light Intensity\n(nW cm-2 sr-1)",
             style = "fixed",
             breaks = breaks,
             palette = "YlOrRd",
             midpoint = NA) +
  tm_layout(main.title = "During Storm (Feb 16)",
            main.title.size = 1.4,
            main.title.position = "center",
            main.title.fontface = "bold",
            legend.outside = FALSE,
            legend.position = c("right", "bottom"),
            legend.bg.color = "white",
            legend.bg.alpha = 0.8,
            legend.frame = TRUE,
            bg.color = "#2b2b2b",
            frame = FALSE) +
  tm_scale_bar(position = c("left", "bottom"),
               text.color = "black",
               bg.color = "white",
               bg.alpha = 0.7) +
  tm_shape(houston_bbox_4326) +
  tm_borders(col = "white", lwd = 1, lty = "dashed") +
  tm_compass(position = c("left", "top"),
             type = "4star",
             size = 2,
             color.dark = "black",
             color.light = "white",
             bg.color = "white",
             bg.alpha = 0.7)

# Display side-by-side with inset map in lower left
tmap_arrange(map_feb07, map_feb16, inset_map,
             ncol = 2, nrow = 2,
             asp = NA,
             outer.margins = 0.02)

```

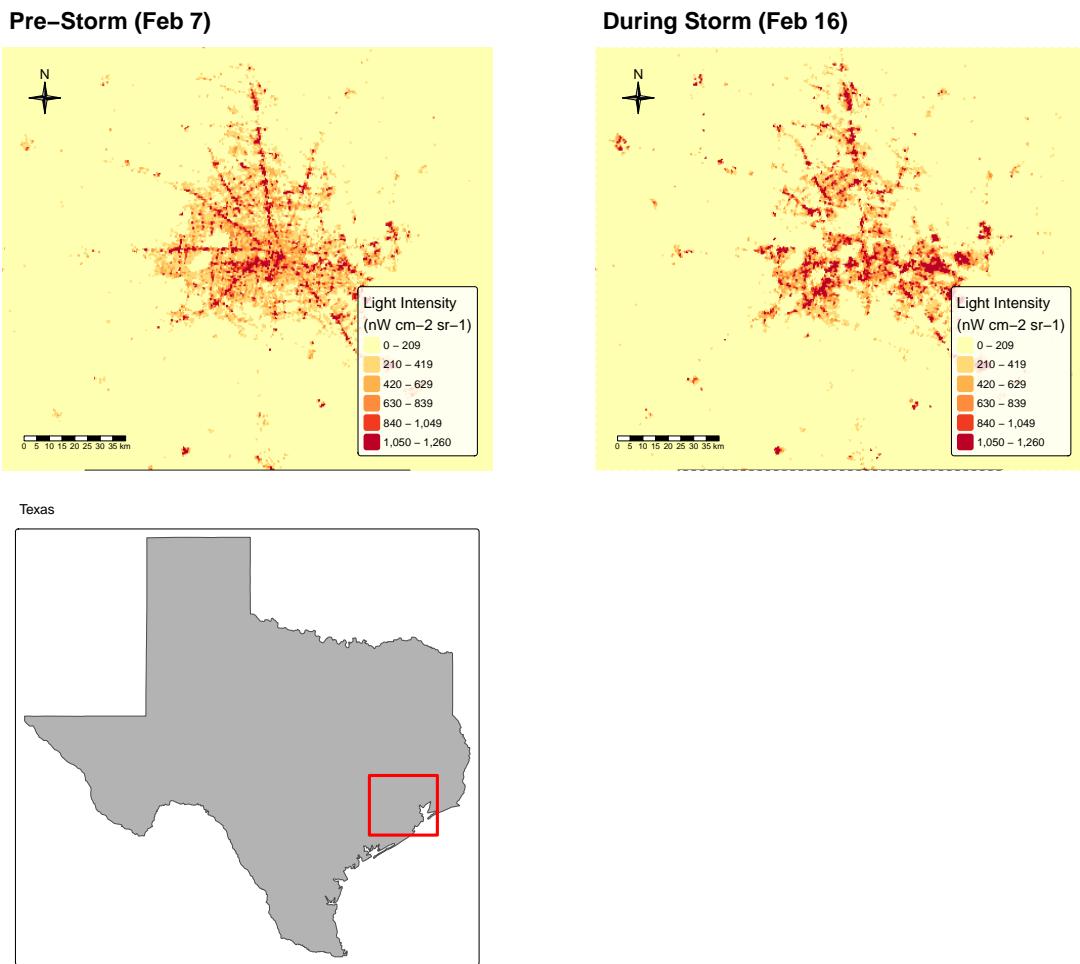


Figure 1: Comparison of night lights in Houston before (February 7, 2021) and during (February 16, 2021) the winter storms showing dramatic reduction in light intensity during the blackout event. Inset map shows Houston's location within Texas.

Exclude Highway Areas

```
# Load highways data from OSM
# First unzip the roads data
unzip("data/gis_osm_roads_free_1.gpkg.zip", exdir = "data")

# Filter to motorways using SQL query
query <- "SELECT * FROM gis_osm_roads_free_1 WHERE fclass='motorway'"
highways <- st_read("data/gis_osm_roads_free_1.gpkg", query = query, quiet = TRUE)
```

```

# Check that highways were loaded
if (nrow(highways) == 0) {
  stop("No motorways found in the OSM roads data!\n",
      "  This could mean:\n",
      "  - The SQL query filter (fclass='motorway') is too restrictive\n",
      "  - The roads file doesn't cover the Houston area\n",
      "  - The data file is corrupted\n",
      "Try checking the available fclass values in the roads data.")
}

cat(" Loaded", nrow(highways), "motorway segments from OSM\n")

```

Loaded 6085 motorway segments from OSM

```

# Reproject to EPSG:3083
highways <- st_transform(highways, crs = 3083)

# Verify CRS matches blackout data
if (st_crs(highways) != st_crs(blackout_houston)) {
  stop("Projection problem! Even after reprojecting to EPSG:3083, the highways don't match b
      " - Highways: ", st_crs(highways)$input, "\n",
      " - Blackout: ", st_crs(blackout_houston)$input, "\n",
      "This is unusual and may indicate a problem with the transformation.")
}

# Create 200m buffer around highways
highway_buffer <- st_buffer(highways, dist = 200)

# Combine all highway buffers into a single geometry
highway_buffer <- st_union(highway_buffer)

# Remove highway areas from blackout mask
# Use st_difference to exclude highway buffer areas
blackout_no_highways <- st_difference(blackout_houston, highway_buffer)

# Check how much area was removed
original_area <- nrow(blackout_houston)
filtered_area <- nrow(blackout_no_highways)
pct_removed <- round((1 - filtered_area/original_area) * 100, 1)
cat("Removed", pct_removed, "% of blackout areas due to highway proximity\n")

```

Removed 0.1 % of blackout areas due to highway proximity

Identify Affected Residential Buildings

```
# Load buildings data from OSM
# First unzip the buildings data
unzip("data/gis_osm_buildings_a_free_1.gpkg.zip", exdir = "data")

# Filter to residential buildings using SQL query
query <- "SELECT * FROM gis_osm_buildings_a_free_1 WHERE (type IS NULL AND name IS NULL) OR"
buildings <- st_read("data/gis_osm_buildings_a_free_1.gpkg", query = query, quiet = TRUE)

# Check that buildings were loaded
if (nrow(buildings) == 0) {
  stop("No residential buildings found in the OSM data!\n",
       "  Possible reasons:\n",
       "  - The SQL query is filtering too strictly\n",
       "  - The buildings file doesn't cover Houston\n",
       "  - The data is missing or corrupted\n",
       "Try examining the 'type' field values in the buildings data to adjust the query.")
}

cat(" Loaded", format(nrow(buildings), big.mark = ","), "residential buildings from OSM\n")
```

Loaded 475,941 residential buildings from OSM

```
# Reproject to EPSG:3083
buildings <- st_transform(buildings, crs = 3083)

# Verify CRS matches blackout data
if (st_crs(buildings) != st_crs(blackout_no_highways)) {
  stop("Cannot identify affected buildings! Projection mismatch detected:\n",
       "  - Buildings: ", st_crs(buildings)$input, "\n",
       "  - Blackout: ", st_crs(blackout_no_highways)$input, "\n",
       "Both datasets must be in EPSG:3083 for accurate spatial intersection.")
}

# Find buildings within blackout areas (spatial join)
# Use st_intersection to find buildings that intersect with blackout areas
affected_buildings <- st_intersection(buildings, blackout_no_highways)
```

```

# Count affected homes
num_affected_homes <- nrow(affected_buildings)

if (num_affected_homes == 0) {
  warning("Surprising result: No residential buildings found in blackout areas!\n",
          "  This could indicate:\n",
          "  - The blackout areas and buildings don't spatially overlap\n",
          "  - All blackout areas were removed by the highway filter\n",
          "  - A projection or data alignment issue\n",
          "Consider visualizing both layers to diagnose the problem.")
} else {
  cat(" Identified", format(num_affected_homes, big.mark = ","), "homes affected by blackout")
  pct_affected <- round((num_affected_homes / nrow(buildings)) * 100, 2)
  cat(" This represents", pct_affected, "% of all residential buildings in the dataset\n")
}

```

Identified 168,874 homes affected by blackout
 This represents 35.48 % of all residential buildings in the dataset

Map: Houston Homes That Lost Power

```

# Create map showing affected residential buildings with enhanced styling
houston_bbox <- st_bbox(blackout_houston)

# Convert affected buildings to centroids to ensure point representation
# (In case st_intersection created complex geometries)
affected_buildings_points <- st_centroid(affected_buildings)

# Layer order: blackout areas (middle) -> red dots (top)
tm_shape(blackout_no_highways) +
  tm_polygons(col = "#2c3e50",
              border.col = "#34495e",
              lwd = 0.3,
              alpha = 0.7) +
  tm_shape(affected_buildings_points) +
  tm_dots(col = "#e74c3c",
          size = 0.03,
          alpha = 0.9,
          shape = 20) +

```

```
tm_layout(main.title = "Houston Residential Buildings That Lost Power",
          main.title.size = 1.5,
          main.title.position = "center",
          main.title.fontface = "bold",
          legend.show = FALSE,
          bg.color = "#ecf0f1",
          frame = TRUE,
          frame.lwd = 2,
          inner.margins = c(0.05, 0.05, 0.05, 0.05)) +
tm_scale_bar(position = c("left", "bottom"),
              text.size = 0.7,
              text.color = "black",
              bg.color = "white",
              bg.alpha = 0.8,
              breaks = c(0, 10, 20)) +
tm_compass(position = c("right", "top"),
            type = "4star",
            size = 2.5,
            text.size = 0.8,
            color.dark = "black",
            color.light = "white",
            bg.color = "white",
            bg.alpha = 0.8)
```

Houston Residential Buildings That Lost Power

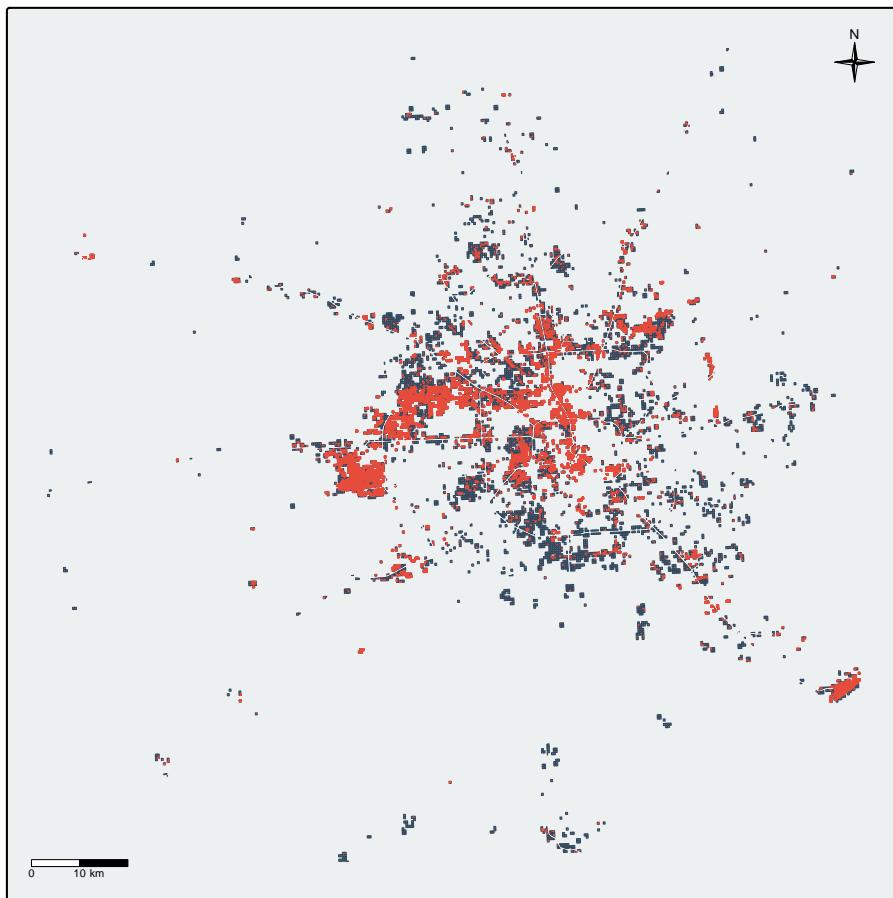


Figure 2: Spatial distribution of residential buildings in Houston that lost power during the February 2021 winter storms. Affected homes shown as bright red points, blackout areas in dark gray.

Analyze Socioeconomic Impacts

Join with Census Data

```
# Load ACS census tract data  
# First unzip the geodatabase  
unzip("data/ACS_2019_5YR_TRACT_48_TEXAS.gdb.zip", exdir = "data")
```

```

# Read the geometry layer
census_tracts <- st_read("data/ACS_2019_5YR_TRACT_48_TEXAS.gdb",
                        layer = "ACS_2019_5YR_TRACT_48_TEXAS",
                        quiet = TRUE)

# Check that census data was loaded
if (nrow(census_tracts) == 0) {
  stop("Failed to load census tract geometries!\n",
       "  Please verify:\n",
       "  - The geodatabase 'ACS_2019_5YR_TRACT_48_TEXAS.gdb' is unzipped in the data folder",
       "  - The layer name 'ACS_2019_5YR_TRACT_48_TEXAS' is correct\n",
       "  Use st_layers() to list available layers in the geodatabase.")
}

cat("  Loaded", format(nrow(census_tracts), big.mark = ","), "census tracts for Texas\n")

```

Loaded 5,265 census tracts for Texas

```

# Read the income data from the X19_INCOME layer (non-spatial table)
income_data <- st_read("data/ACS_2019_5YR_TRACT_48_TEXAS.gdb",
                        layer = "X19_INCOME",
                        quiet = TRUE)

# Check that income data was loaded
if (nrow(income_data) == 0) {
  stop("Failed to load income data from the geodatabase!\n",
       "  Please verify:\n",
       "  - The layer 'X19_INCOME' exists in the geodatabase\n",
       "  - The geodatabase is not corrupted\n",
       "  Use st_layers() to check available layers.")
}

# Join income data with geometry based on GEOID
# The income layer uses just "GEOID" while geometry layer has "GEOID_Data"
# We need to match GEOID from geometry with GEOID from income data
census_tracts_original <- nrow(census_tracts)
census_tracts <- census_tracts %>%
  left_join(st_drop_geometry(income_data), by = c("GEOID_Data" = "GEOID")) %>%
  select(GEOID, B19013e1) %>%
  rename(median_income = B19013e1) %>%
  # Filter out tracts with missing or zero income

```

```

filter(!is.na(median_income) & median_income > 0)

# Warn if many tracts were removed
tracts_removed <- census_tracts_original - nrow(census_tracts)
pct_removed <- round((tracts_removed / census_tracts_original) * 100, 1)
if (pct_removed > 10) {
  warning("Heads up! ", pct_removed, "% of census tracts were removed due to missing income data",
         " This may affect the representativeness of your socioeconomic analysis.\n",
         " Missing data could be systematic (e.g., certain types of tracts underreported).")
}

cat(" Retained", format(nrow(census_tracts), big.mark = ","), "of",
    format(census_tracts_original, big.mark = ","), "census tracts with valid income data\n")

```

Retained 5,202 of 5,265 census tracts with valid income data

```

# Reproject to EPSG:3083
census_tracts <- st_transform(census_tracts, crs = 3083)

# Verify CRS matches blackout data
if (st_crs(census_tracts) != st_crs(blackout_no_highways)) {
  stop("Cannot analyze socioeconomic impacts! Projection mismatch between census and blackout areas",
       " - Census: ", st_crs(census_tracts)$input, "\n",
       " - Blackout: ", st_crs(blackout_no_highways)$input, "\n",
       "Both must be in EPSG:3083 for accurate spatial analysis.")
}

# Spatially join census tracts with blackout areas
# First, find which census tracts intersect with blackout areas
affected_tracts <- st_join(census_tracts, blackout_no_highways, left = FALSE)

# Get unique tract IDs that were affected
affected_tract_ids <- unique(affected_tracts$GEOID)

# Check if any tracts were affected
if (length(affected_tract_ids) == 0) {
  warning("Unexpected result: No census tracts intersect with blackout areas!\n",
         " This suggests a spatial alignment problem. Please check:\n",
         " - Both datasets are in the same projection (EPSG:3083)\n",
         " - The blackout areas actually cover Houston (where census tracts exist)\n",
         " - The spatial join is using the correct method\n",
         )
}

```

```

        "Try visualizing both layers together to diagnose the issue.")
} else {
  cat(" Identified", length(affected_tract_ids), "census tracts affected by blackouts\n")
}

Identified 927 census tracts affected by blackouts

# Categorize tracts as affected or unaffected
census_tracts <- census_tracts %>%
  mutate(blackout_status = ifelse(GEOID %in% affected_tract_ids,
                                  "Affected",
                                  "Not Affected"))

# Filter out tracts with missing income data
census_tracts <- census_tracts %>%
  filter(!is.na(median_income) & median_income > 0)

# Summary of categorization
cat("\nBlackout status summary:\n")

```

Blackout status summary:

```
print(table(census_tracts$blackout_status))
```

| Affected | Not Affected |
|----------|--------------|
| 927 | 4275 |

Map: Blackout Events by Census Tract

```

# Create map showing which census tracts experienced blackouts
# Crop to Houston area for better visualization
houston_bbox <- st_bbox(blackout_houston)
census_houston <- st_crop(census_tracts, houston_bbox)

tm_shape(census_houston) +
  tm_polygons(col = "blackout_status",

```

```

palette = c("Affected" = "coral", "Not Affected" = "lightblue"),
title = "Blackout Status",
border.col = "white",
lwd = 0.5) +
tm_layout(main.title = "Census Tracts Experiencing Blackouts",
main.title.size = 1.3,
main.title.position = "center",
legend.outside = TRUE,
legend.outside.position = "right",
frame = TRUE,
bg.color = "#ecf0f1") +
tm_scale_bar(position = c("left", "bottom")) +
tm_compass(position = c("right", "top"))

```

Census Tracts Experiencing Blackouts

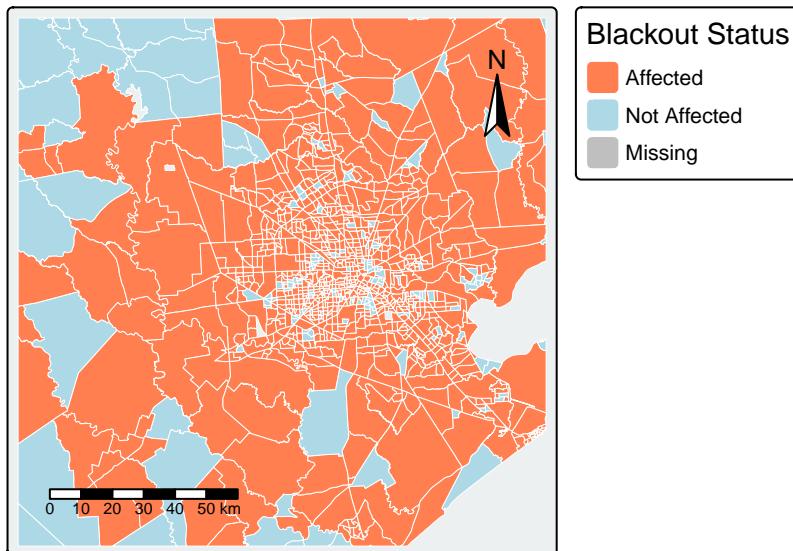


Figure 3: Census tracts in Houston classified by blackout status, with affected tracts shown in coral and unaffected tracts in light blue.

Income Distribution Analysis

```

# Create plot comparing median household income distribution
# for census tracts that experienced blackouts vs those that didn't

# Filter to Houston area census tracts only
census_houston_data <- census_houston %>%
  st_drop_geometry()

# Create distribution plot
ggplot(census_houston_data, aes(x = median_income, fill = blackout_status)) +
  geom_histogram(position = "dodge", bins = 30, alpha = 0.7) +
  scale_fill_manual(values = c("Affected" = "coral", "Not Affected" = "lightblue"),
                     name = "Blackout Status") +
  scale_x_continuous(labels = scales::dollar_format()) +
  labs(title = "Distribution of Median Household Income by Blackout Status",
       subtitle = "Houston Census Tracts, 2021 Winter Storms",
       x = "Median Household Income (2019 ACS)",
       y = "Number of Census Tracts") +
  theme_minimal() +
  theme(plot.title = element_text(size = 14, face = "bold"),
        plot.subtitle = element_text(size = 11),
        legend.position = "bottom")

```

Distribution of Median Household Income by Blackout

Houston Census Tracts, 2021 Winter Storms

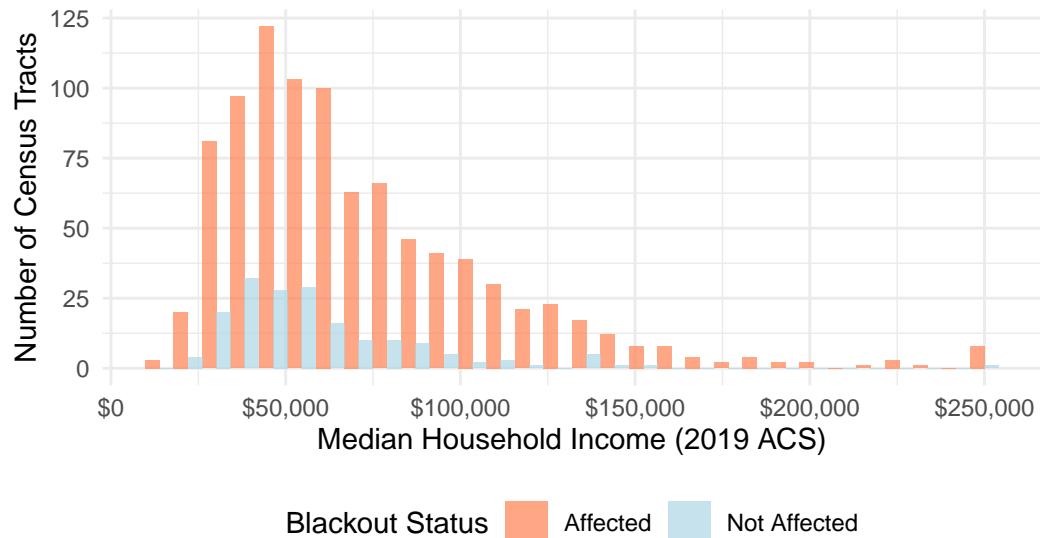


Figure 4: Distribution of median household income across Houston census tracts, comparing areas that experienced blackouts (coral) versus those that did not (light blue)

Results Summary

```
# Calculate and display key statistics:  
# - Number of homes affected  
# - Number of census tracts affected  
# - Summary statistics of income distributions  
  
# Number of homes affected  
cat("Total number of residential buildings affected:", num_affected_homes, "\n\n")
```

Total number of residential buildings affected: 168874

```
# Number of census tracts affected vs not affected  
tract_summary <- census_houston_data %>%  
  count(blackout_status) %>%  
  rename(`Blackout Status` = blackout_status, `Number of Tracts` = n)  
  
print(tract_summary)
```

```

Blackout Status Number of Tracts
1      Affected          927
2 Not Affected         177

cat("\n")
# Income statistics by blackout status
income_summary <- census_houston_data %>%
  group_by(blackout_status) %>%
  summarise(
    `Mean Income` = scales::dollar(mean(median_income, na.rm = TRUE)),
    `Median Income` = scales::dollar(median(median_income, na.rm = TRUE)),
    `Min Income` = scales::dollar(min(median_income, na.rm = TRUE)),
    `Max Income` = scales::dollar(max(median_income, na.rm = TRUE)),
    `Std Dev` = scales::dollar(sd(median_income, na.rm = TRUE))
  ) %>%
  rename(`Blackout Status` = blackout_status)

knitr::kable(income_summary,
             caption = "Median Household Income Statistics by Blackout Status")

```

Table 1: Summary statistics of median household income for census tracts by blackout status

Table 1: Median Household Income Statistics by Blackout Status

| Blackout Status | Mean Income | Median Income | Min Income | Max Income | Std Dev |
|-----------------|-------------|---------------|------------|------------|-------------|
| Affected | \$72,095.42 | \$61,188 | \$13,886 | \$250,001 | \$39,605.27 |
| Not Affected | \$59,057.65 | \$51,583 | \$24,024 | \$250,001 | \$29,969.30 |

Discussion

This analysis identified **168,874 residential buildings (35.5% of Houston homes)** that **lost power** during the February 2021 winter storms, affecting **927 of 1,104 census tracts (84%)**. Surprisingly, affected tracts had higher median incomes (\$61,188) than unaffected areas (\$51,583), contradicting typical environmental justice patterns. This may reflect Houston's settlement geography. The analysis demonstrates remote sensing's value for rapid disaster assessment—VIIRS data enabled impact quantification within days without ground surveys. However, night lights imperfectly correlate with residential power loss, the 200m highway buffer is arbitrary, 2019 income data may not reflect 2021 conditions, and tract-level aggregation masks within-neighborhood variation.

Key Findings

- 168,874 homes affected (35.5% of residential buildings)
- 84% of census tracts experienced blackouts
- Counter-intuitively, higher-income areas (\$9,600 median income difference) showed greater blackout rates

Limitations

- Night lights may not perfectly correlate with residential power outages
- Highway exclusion buffer (200m) is arbitrary
- Census tract aggregation masks within-tract variation
- Two-year gap between 2019 ACS data and 2021 storm
- OSM building classification may be incomplete