EDA Proposal

Akash Sivasubramanian - 0862944

Importing the required libraries

```
Rows: 10,992
Columns: 10
$ city           <chr> "Ada", "Addison", "Adrian", "Adrian", "Albion", "Albion~
$ country        <chr> "United States", "United States", "United States", "Uni~
$ description    <chr> "Ada witch - Sometimes you can see a misty blue figure ~
$ location       <chr> "Ada Cemetery", "North Adams Rd.", "Ghost Trestle", "Si~
$ state          <chr> "Michigan", "Michigan", "Michigan", "Michigan", "Michig~
$ state_abbrev   <chr> "MI", "MI", "MI", "MI", "MI", "MI", "MI", "MI", "MI", "~
$ longitude      <dbl> -85.50489, -84.38184, -84.03566, -84.01757, -84.74518, ~
$ latitude       <dbl> 42.96211, 41.97142, 41.90454, 41.90571, 42.24401, 42.23~
$ city_longitude <dbl> -85.49548, -84.34717, -84.03717, -84.03717, -84.75303, ~
$ city_latitude  <dbl> 42.96073, 41.98643, 41.89755, 41.89755, 42.24310, 42.24~
```

Wow there are 9904 unique locations in 4386 unique cities across usa. This is interesting. Hmm things going weired because there are only 50 states in the US but we have 51 unique values in the state column. Let's dig deeper into this.

```
# A tibble: 1 x 8
   city description location state latitude longitude city_latitude
  <int>       <int>    <int> <int>    <int>     <int>         <int>
1     3           0        3     0     1261      1261            29
# i 1 more variable: city_longitude <int>
```

printing the rows with missing values in city_latitude and city_longitude columns.

creating a frequency table for the city ,location and state columns

Table 1: A table of City Frequency and Proportion

City	Counts	Proportions
Los Angeles	61	0.0139844
San Antonio	55	0.0126089
Honolulu	43	0.0098579
Pittsburgh	42	0.0096286
Columbus	41	0.0093994

Table 2: A table of State Frequency and Proportion

State	Counts	Proportions
California	1067	20.921569
Texas	696	13.647059
Pennsylvania	648	12.705882
Michigan	526	10.313726
Ohio	475	9.313726

Create a proportion table for it.

Ok now we can see that some of the location names have some additional data and typos and with different cases. Let's clean this. we have total of 638 rows of the cities with more that one common lat and long.

Super. it worked . Now we can see that the location names are cleaned.

Now we can apply the same method for entire dataframe.

Yeah it worked. Now we can see that the location names are cleaned.

Table 3: A table of Location with multiple citing

Location	Counts
Prince Georges county	20
Cry Baby Bridge	14
Cemetery	13
Mission Inn	12
Oviedo	12

this is the true location frequency.

these are the locations with multiple ghost citings. lets plot it on the map.

get top 25 places with multiple haunted citing.

Modelling

We can analyze if there's a relationship between distance from city center and the concentration of haunted places. we can create a multilinear regression model to predict the number of haunted places in a city based on the distance from the city center.

State Distribution

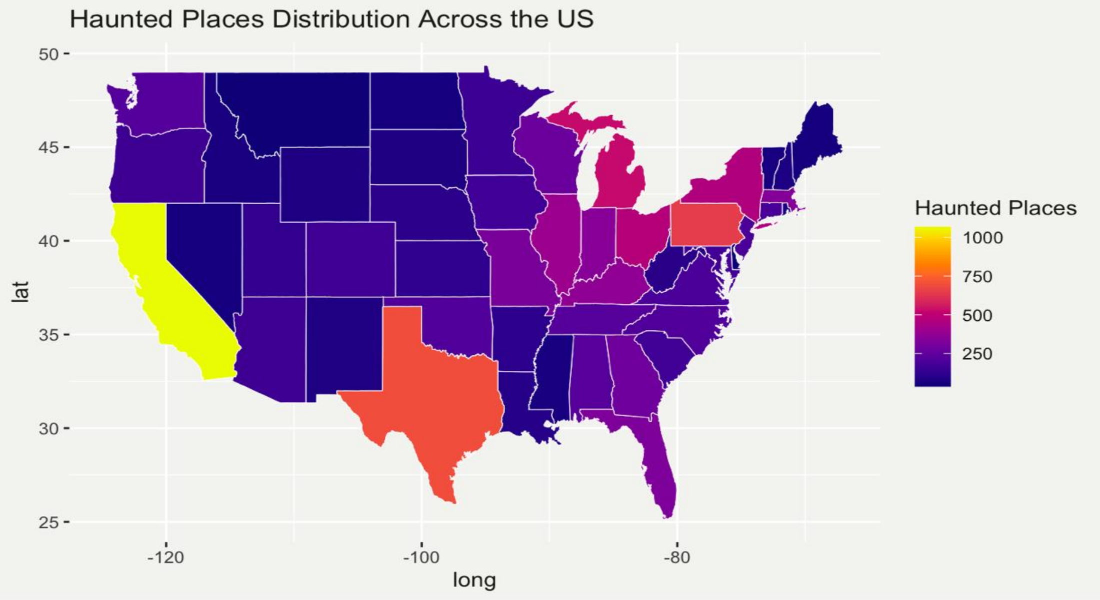


Figure 1: Choropleth Map of states

Bubble map of top 50 haunted places

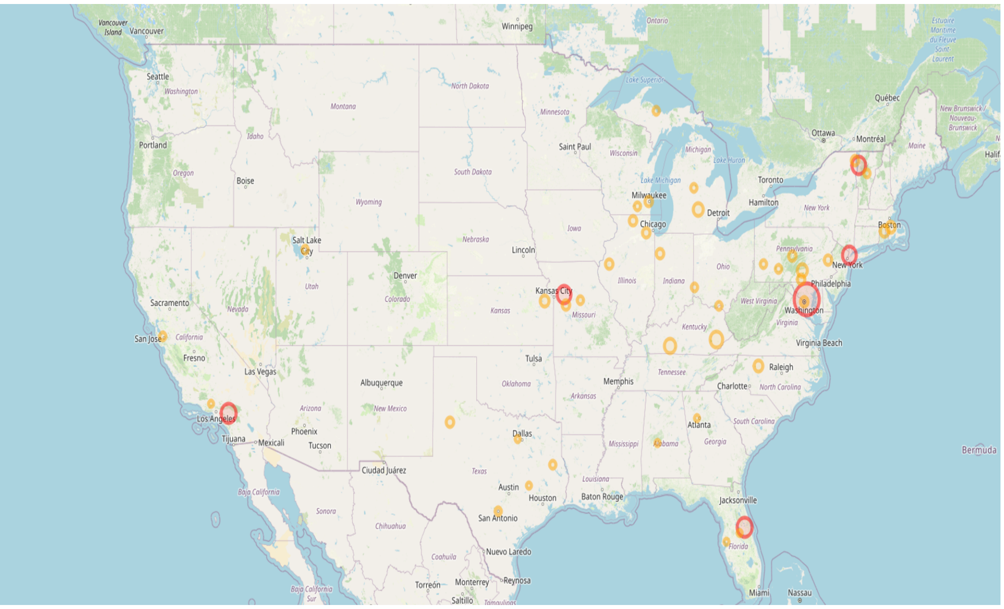


Figure 2: Bubble Map

Word Cloud of the cities

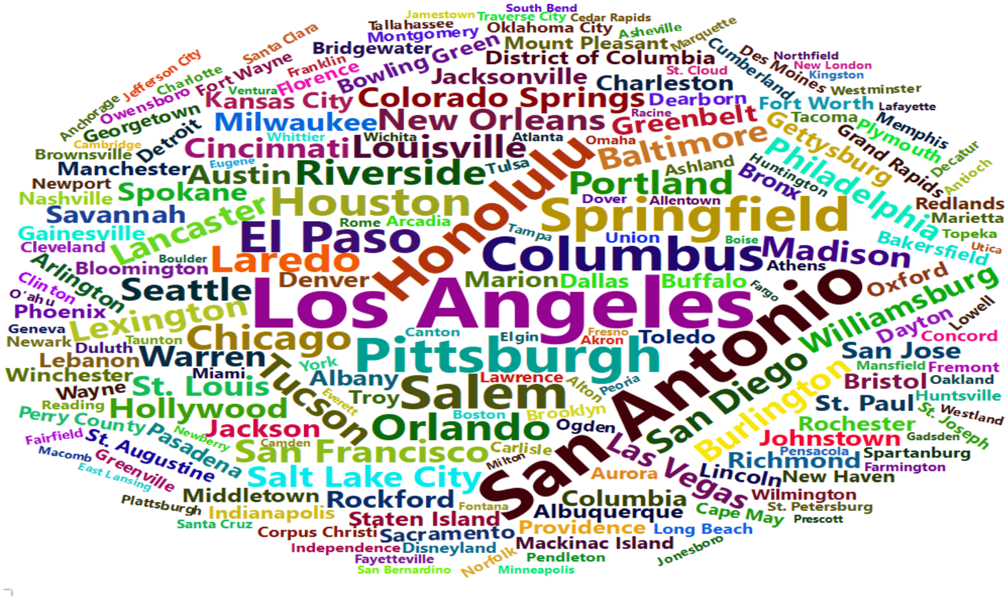


Figure 3: Word Cloud of Cities

Bar plot of Top 20 hauntred siting in cities

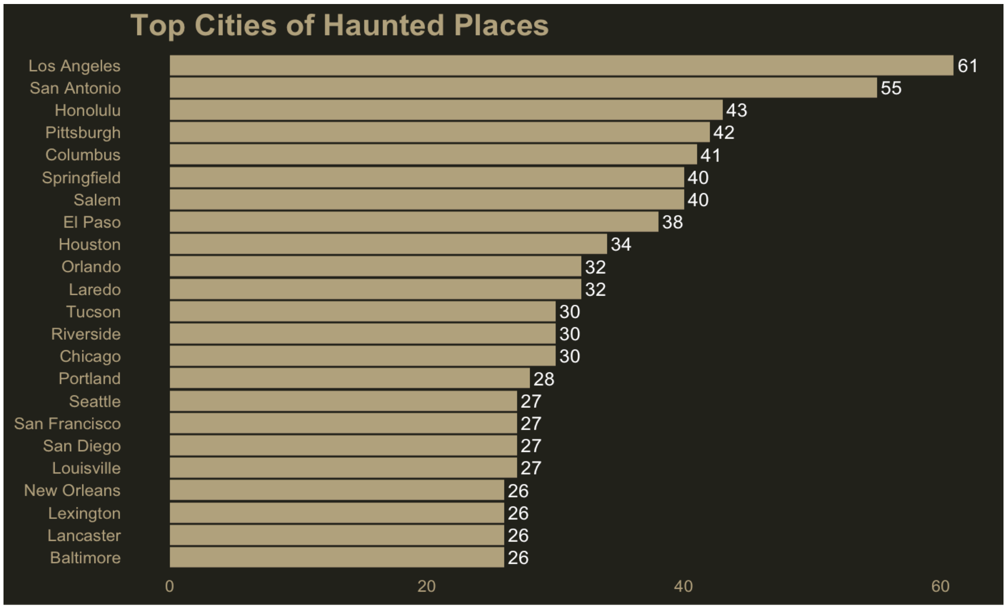


Figure 4: Bar plot of Top 20 hauntred siting in cities

Null Hypothesis H_0 : proportion of simulation rejections found using the CLT based approach was equal to 10%.

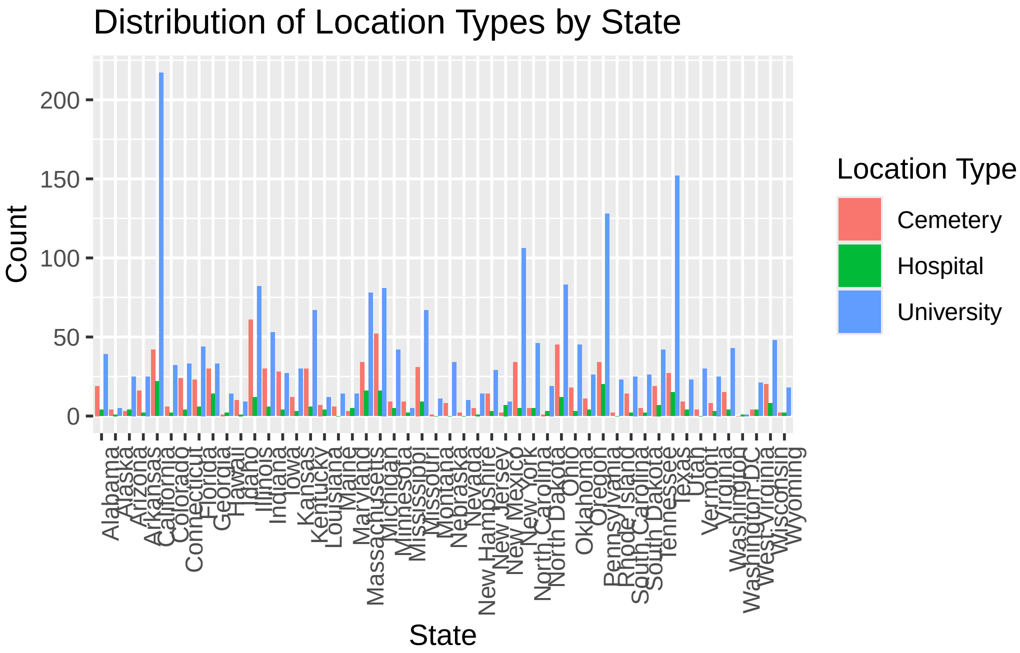
$$H_0 : p = \text{Time in Bed (TIB)}$$

Alternative Hypotheses H_A : proportion of simulation rejections found using the CLT based approach was different from 10%.

$$H_1 : p \neq \text{Time in Bed (TIB)}$$

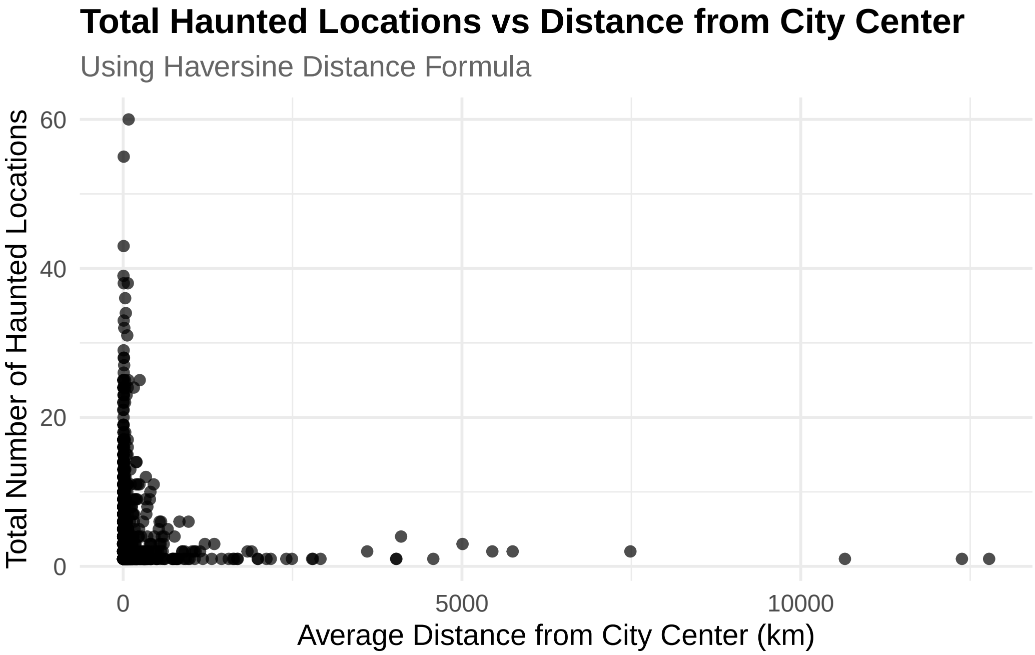
```
Warning in chisq.test(contingency_table): Chi-squared approximation may be
incorrect
```

```

    Pearson's Chi-squared test

data:  contingency_table
X-squared = 252.9, df = 98, p-value = 1.218e-15
```



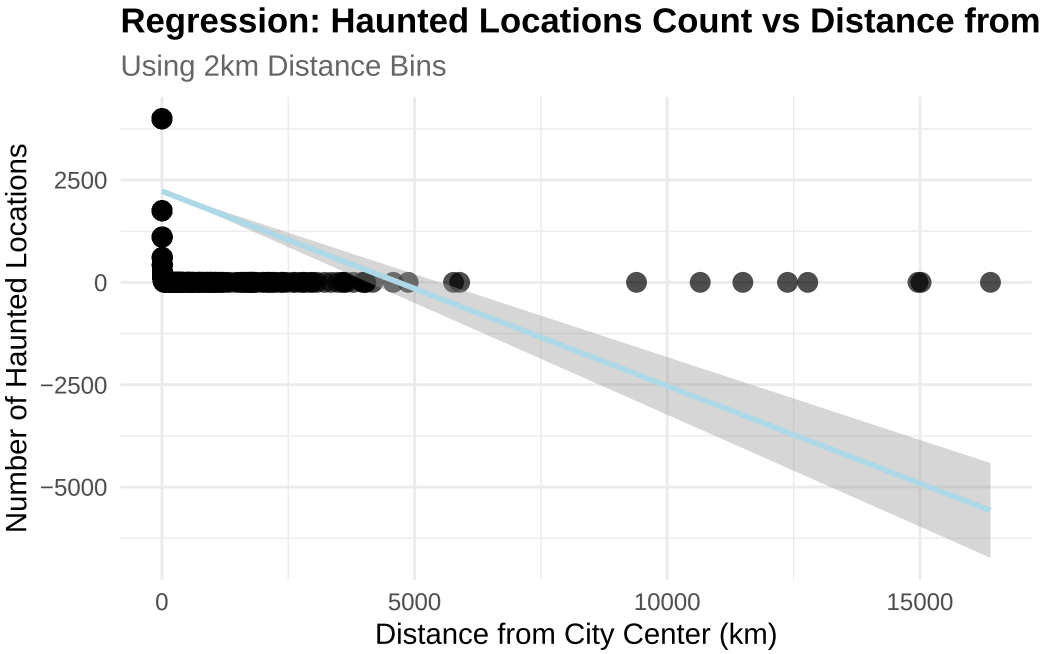
```
# A tibble: 9,718 x 7
   city    location        state latitude longitude city_latitude city_longitude
   <chr>   <chr>           <chr>    <dbl>     <dbl>         <dbl>          <dbl>
 1 Ada     Ada Cemetery    Mich~     43.0     -85.5          43.0          -85.5
 2 Addison North Adams Rd. Mich~     42.0     -84.4          42.0          -84.3
 3 Adrian  Ghost Trestle   Mich~     41.9     -84.0          41.9          -84.0
 4 Adrian  Siena Heights ~ Mich~     41.9     -84.0          41.9          -84.0
 5 Albion  Albion College  Mich~     42.2     -84.7          42.2          -84.8
 6 Albion  Riverside Ceme~ Mich~     42.2     -84.8          42.2          -84.8
 7 Algonac Morrow Road     Mich~     42.7     -82.6          42.6          -82.5
 8 Allegan Elks Lodge      Mich~     42.5     -85.8          42.5          -85.9
 9 Allegan The Grill Hous~ Mich~     42.5     -85.9          42.5          -85.9
10 Allegan The Yellow Mot~ Mich~     42.5     -85.9          42.5          -85.9
# i 9,708 more rows
```

```
Rows: 9,718
Columns: 8
$ city           <chr> "Ada", "Addison", "Adrian", "Adrian", "Albion", "Al~
$ location       <chr> "Ada Cemetery", "North Adams Rd.", "Ghost Trestle",~
$ state          <chr> "Michigan", "Michigan", "Michigan", "Michigan", "Mi~
$ latitude       <dbl> 42.96211, 41.97142, 41.90454, 41.90571, 42.24401, 4~
$ longitude      <dbl> -85.50489, -84.38184, -84.03566, -84.01757, -84.745~
$ city_latitude      <dbl> 42.96073, 41.98643, 41.89755, 41.89755, 42.24310, 4~
$ city_longitude     <dbl> -85.49548, -84.34717, -84.03717, -84.03717, -84.753~
$ distance_from_city <dbl> 0.78115583, 3.31676740, 0.78733008, 1.85898100, 0.6~
```



```
Call:
lm(formula = total_haunted ~ avg_distance, data = binned_data)

Residuals:
   Min     1Q Median     3Q    Max 
 -2203  -1618   -476   1771   5575 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2228.41466   16.44421  135.51   <2e-16 ***
avg_distance   -0.47588    0.03611  -13.18   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1596 on 9515 degrees of freedom
  (201 observations deleted due to missingness)
Multiple R-squared:  0.01793,   Adjusted R-squared:  0.01783 
F-statistic: 173.7 on 1 and 9515 DF,  p-value: < 2.2e-16
```

```
`geom_smooth()` using formula = 'y ~ x'

Warning: Removed 201 rows containing non-finite outside the scale range
(`stat_smooth()`).

Warning: Removed 201 rows containing missing values or values outside the scale range
(`geom_point()`).
```



Methodology

What are the questions we are going to answer? 1 Are certain types of haunted locations (e.g., cemeteries, schools, universities) more common in specific states? 2 How does the distance of haunted locations from city centers relate to the total number of haunted locations in a city? ### How to Answer the questions & Why ? - *Chi-Square Test of Independence* was appropriate for analyzing categorical relationships between location types and states. - *Linear Regression* was suitable for examining numerical relationships, particularly between distance from city centers and the number of haunted locations.

Assumptions

- For the chi-square test, it was assumed that the data is random and that each observation is independent, and the each variable has more than 5 observations
- For regression analysis, assumptions included linearity, independence of errors, homoscedasticity, and normality of residuals. Diagnostic checks were conducted to validate these assumptions.

Technical Details

Analytical Process

1. *Data Cleaning:*

   - Removed redundant columns such as country and standardized location names.
   - Addressed missing values by imputing coordinates or filtering incomplete rows.
   - Eliminated duplicate rows to ensure data integrity.

2. *Exploratory Data Analysis (EDA):*

   - Summarized key variables and generated visualizations (e.g., word clouds, bar plots, and maps).

3. *Chi-Square Test:*

   Null Hypothesis H_0 : Distribution of location types (cementry, university and hospital) are independent of the state

Alternative Hypotheses H_A : Distribution of location types (cementry, university and hospital) are dependent of the state.

- Created a contingency table to test the relationship between location type and state.
- Conducted the test using the chisq.test function in R.

5. *Linear Regression Modeling:*

$$\text{Number of Haunted Places} = \beta_0 + \beta_1 \times \text{Average Distance from City Center}$$

   - Calculated distances of haunted locations from city centers.
   - Aggregated data by city to compute average distances and total haunted locations by using **Haversine distance formula**.
   - Built a regression model using the lm() function in R.

Parameters and Thresholds

- For chi-square tests, a significance level of 0.05 was used to determine statistical significance. #### Data Transformations
- Calculated distance_from_center as the Haversine distance between latitude and longitude of the haunted location and the city center. (In the presentation we used Euclidean distance but realized that it is not accurate, when dealing with curvilinear surfaces)
- Standardized location names to address inconsistencies and typos.
- Binned distances into 2km intervals to analyze the relationship between distance from city center and the number of haunted locations.

Tools and Packages

The following R packages were utilized: - **dplyr and tidyr:** For data cleaning and manipulation. - **ggplot2:** For creating visualizations. - **knitr and kableExtra:** For generating tables in the report. - **leaflet and wordcloud2:** For interactive maps and word clouds. - **regular expression:** for word processing, filtering and searching.

Analysis

Results Presentation

1. Location Type and State Dependency (Chi-Square Test of Independence)

To assess whether the distribution of haunted location types (e.g., cemeteries, universities, hospitals) varies significantly across states, a chi-square test of independence was conducted.

- **Test Results:**
  - **Chi-Square Statistic**: 252.9
  - **Degrees of Freedom (df)**: 98
  - **p-value**: $\boxed{1.218\times\ 10^{-15}}$
- **Key Findings:**
  - The chi-square test yielded a highly significant result ((p < 0.05)), indicating a strong relationship between the type of haunted location and the state it is located in.
  - The observed dependence suggests that the types of haunted locations (e.g., cemeteries, universities) are not evenly distributed across states. This may reflect regional differences in cultural, historical, or geographic factors that influence the prevalence of certain haunted locations.

- **Visual Summary:**

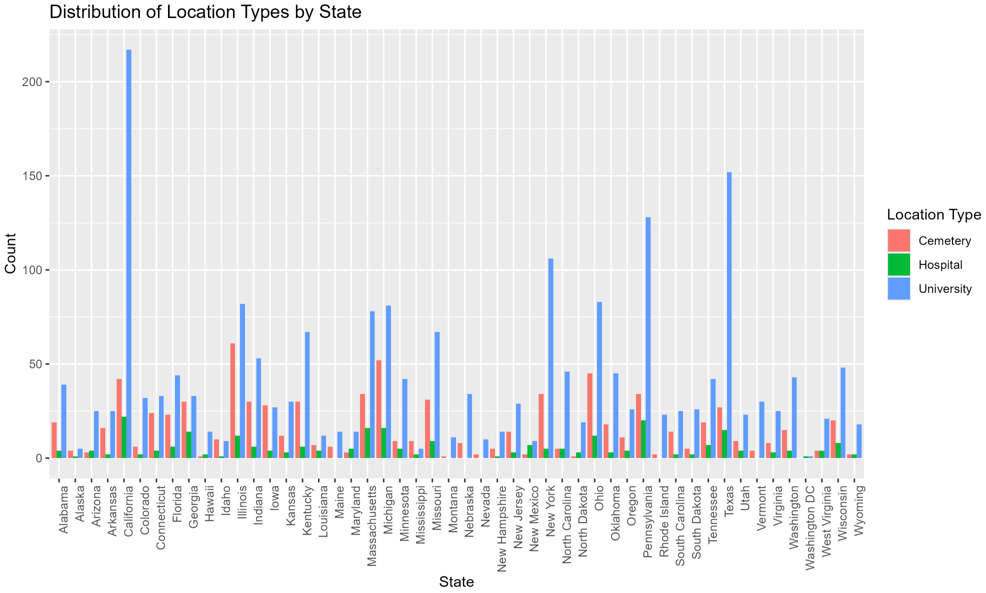


Figure 5: Distribution of haunted location

- A bar plot was used to display the contingency table, visually highlighting the differences in the distribution of haunted location types across states.
- **Interpretation:**
  - These findings suggest that specific states may have more of certain haunted location types due to unique regional factors. For example, a state with a rich history of educational institutions may exhibit a higher prevalence of haunted universities. Similarly, regions with older, historic graveyards may report more haunted cemeteries.

2. Relationship Between Distance from City Centers and Number of Haunted Locations (Linear Regression)

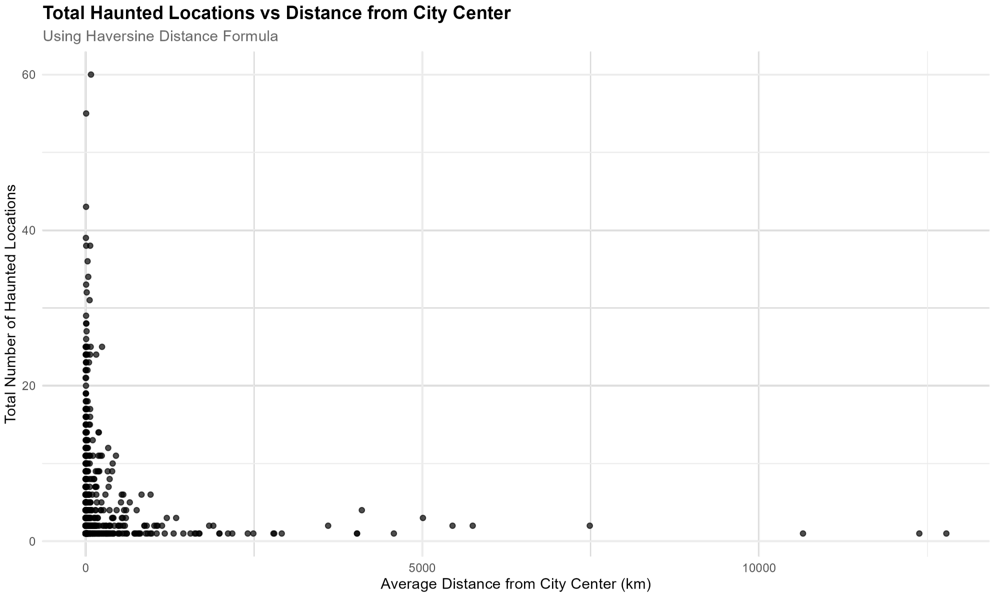


Figure 6: Scatter plot of the location count with respect to distance from city center

We can clearly see that the number of haunted locations are more near to the city center. A linear regression analysis was performed to explore whether the average distance of haunted locations from city centers predicts the total number of haunted locations in a city. The regression model is expressed as:

```
$$
\text{Number of Haunted Places} = \beta_0 + \beta_1 \times \text{Average Distance from City
$$
```

- **Model Summary:**
  - **Intercept** β_0: (2228.41 ±16.44), (t = 135.51), ($p < 2 \times\ 10^{-16}$)
  - **Slope** β_1: (-0.4759 ±0.0361), (t = -13.18), ($p < 2 \times\ 10^{-16}$)
  - **Residual Standard Error**: 1596
  - **R-squared** R^2: (0.0179) (1.79% of variance explained)
- **Key Findings:**
  - The negative coefficient ($\beta_1\ <\ 0$) indicates an inverse relationship between the average distance of haunted locations from city centers and the total number of haunted places.

  - This suggests that cities with haunted locations closer to their centers tend to have more haunted places overall.
  - However, the low (R^2) value indicates that only 1.79% of the variance in the number of haunted locations is explained by distance, implying that additional factors are likely contributing to this relationship.

- **Visual Summary:**
  - A **scatter plot** with a fitted regression line was created to illustrate the negative relationship between distance and the number of haunted places.
  - **Residual diagnostics** were conducted:
    - **Residuals vs. Fitted**: Confirmed the linearity of the model.
    - **Q-Q Plot**: Showed non normality of residuals.
    - **Scale-Location Plot**: Suggested homoscedasticity of residuals.
    - **Residuals vs. Leverage**: Highlighted potential outliers influencing the model.

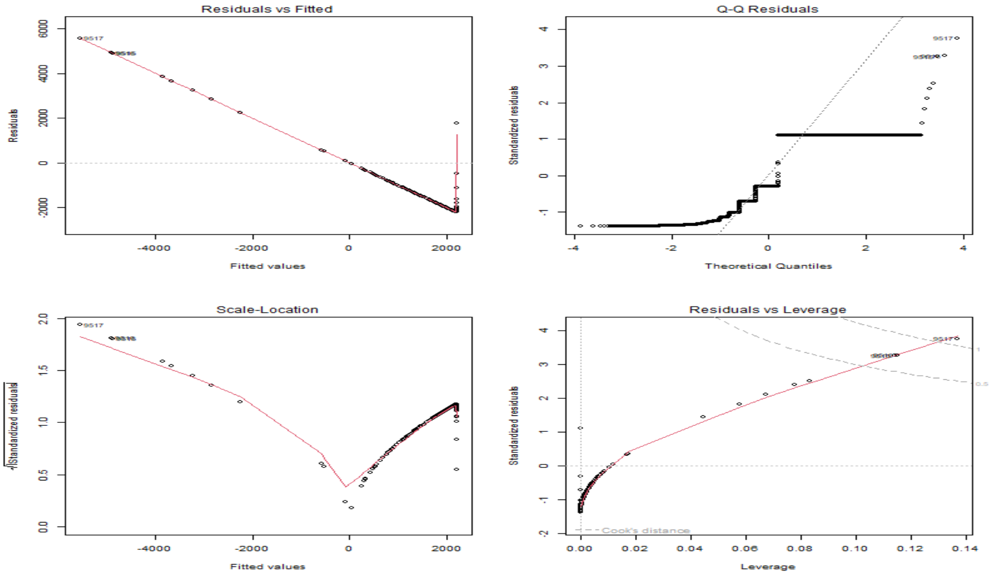


Figure 7: Diagnostic plots

- **Interpretation:**
  - While the negative relationship between distance and the number of haunted places is statistically significant, the low explanatory power R^2 suggests that this variable alone is insufficient to explain the total number of haunted locations.

    - Other variables, such as population density, historical significance, or urbanization levels, may play a significant role in determining the distribution of haunted places.

Quality Checks and Limitations

1. **Assumption Checks for Chi-Square Test:**
    - The minimum expected frequency condition for the chi-square test was met, ensuring the validity of the test.
2. **Regression Diagnostics:**
    - **Linearity**: The relationship between the predictors and the outcome variable was approximately linear.

    - **Homoscedasticity**: Residual plots suggested constant variance.

    - **Normality**: Q-Q plots indicated that the residuals were approximately normally distributed.
3. **Limitations:**
    - The regression model has a low (R^2), indicating that most of the variance in the number of haunted locations remains unexplained. Additional predictors could enhance the model.
    - Potential outliers in the data might influence the regression results and require further examination.
    - Chi-square results, while significant, do not quantify the strength of the relationship or provide causal insights.
4. **Decisions Taken:**
    - Focused on average distance as the primary predictor to simplify the analysis while maintaining interpretability.
    - Retained significant results but acknowledged limitations in predictive power and model fit.

Summary of Findings

- The chi-square test confirmed a significant association between location types and states, highlighting regional variations in haunted locations.
- The linear regression model revealed a statistically significant, albeit weak, negative relationship between distance from city centers and the number of haunted locations, suggesting that proximity to urban centers may slightly influence the prevalence of haunted places.

Enrire Code

```
library(tidyverse)
library(leaflet)
library(wordcloud2)
library(webshot)
library("htmlwidgets")
library(knitr)
webshot::install_phantomjs()
# Importing the dataset
tuesdata <- tidytuesdayR::tt_load('2023-10-10')
haunted_places <- tuesdata$haunted_places

# Glimpse of the dataset
haunted_places %>% glimpse()
# Counting unique values in each column
column_summary <- haunted_places %>%
  summarise(across(everything(), ~ n_distinct(.)))
column_summary

# Get unique values in the state column in alphabetical order.
unique_states <- haunted_places %>%
  distinct(state) %>%
  arrange(state)

unique_states

# Gottcha there is a value 'Washington D.C.' which is not a state
# but a federal district. Let's correct this.
# we can dig deeper into the dataset to find out the
# haunted places in Washington D.C.
# Haunted places in Washington D.C.
haunted_places_dc <- haunted_places %>%
  filter(state == "Washington DC")
haunted_places_dc

# OK Ok. I think there is lot going on in the Us.
# "After some web search I found that the Wasington DC is a separate entity
# from the US but overseen by US"
# Ohh, That's there political concern. Iam going to keep it as it is.
# Cleaning the dataset
```

```r
# Removing redundant columns (Country, state_abbrev)
haunted_places <- haunted_places %>%
  select(city,
         description,
         location,
         state,
         latitude,
         longitude,
         city_latitude,
         city_longitude)
# Function for checking missing values
missing_values <- function(df){

  df %>% summarise(across(everything(), ~ sum(is.na(.))))
}
missing_values(haunted_places)
# Print rows with missing values in city_latitude and city_longitude columns
missing_coordinates <- haunted_places %>%
  filter(is.na(city_latitude) | is.na(city_longitude))
missing_coordinates
for (city in missing_coordinates$city) {
  print(haunted_places[haunted_places$city == city, ])
}

# by doing the city search we can see that we can find the city latitude and
# longitude of these cities.
# Cockeysville, Faribault, Streamwood, Cynthiana
# Lets fix these city coordinates. (may be later)
# Dig deeper into the missing values
# removing the rows with missing values in city, location,
# city_latitude, city_longitude columns
haunted_places <- haunted_places %>%
  filter(!is.na(city) &
           !is.na(location) &
           !is.na(city_latitude) &
           !is.na(city_longitude))

# Checking for missing values
missing_values(haunted_places)
# Before removing the duplicates
nrow(haunted_places)
```

```r
# Find exact duplicates
duplicate_rows <- haunted_places %>%
  group_by(across(everything())) %>%
  filter(n() > 1) %>%
  ungroup()

# Display the duplicate rows
print(duplicate_rows)

# Remove the Exact duplicate rows
haunted_places <- haunted_places %>%
  distinct()

# find the total number of rows in the dataset after removing the duplicates
nrow(haunted_places)


# Frequency table for city column
city_freq <- haunted_places %>%
  count(city, sort = TRUE)

# Create a proportion table for the location column
city_prop <- city_freq %>%
  mutate(proportion = n / 4362)

kable(city_prop[1:5,], caption = "Frequency and Proportional Table of City",
      col.names = c("City", "Counts", "Proportions"))


# Frequency table for state column
state_freq <- haunted_places %>%
  count(state, sort = TRUE)

# Create a proportion table for the location column
city_prop <- state_freq %>%
  mutate(proportion = n / 51)

kable(city_prop[1:5,], caption = "Freq and Prop Table of City",
      col.names = c("State", "Counts", "Proportions"))

# wordcloud for city column
city_cloud <- wordcloud2(city_freq, size = 0.5)
```

```r
city_cloud

# wordcloud for location column
location_cloud <- wordcloud2(location_freq, size = 0.5)
location_cloud

# wordcloud for state column
state_cloud <- wordcloud2(state_freq, size = 0.8)
state_cloud

# Converting the the wordcloud into image
# save it in html
saveWidget(city_cloud,"tmp_city.html",selfcontained = F)
saveWidget(state_cloud,"tmp_state.html",selfcontained = F)
saveWidget(location_cloud,"tmp_location.html",selfcontained = F)

# and in png or pdf
webshot("tmp_city.html","city_cloud.png", delay =5)
webshot("tmp_state.html","state_cloud.png", delay =5)
webshot("tmp_location.html","location_cloud.png", delay =5)
# Filter rows where 'location' contains "Cemetery"
cemetery_coordinates <- haunted_places %>%
  filter(grepl("cemetery", location, ignore.case = TRUE)) %>%
  select(location, latitude, longitude) %>%
  arrange(location)

school_coordinates <- haunted_places %>%
  filter(grepl("school", location, ignore.case = TRUE)) %>%
  select(location, latitude, longitude) %>%
  arrange(location)

university_coordinates <- haunted_places %>%
  filter(grepl("university", location, ignore.case = TRUE)) %>%
  select(location, latitude, longitude) %>%
  arrange(location)

# Print the result
cemetery_coordinates
school_coordinates
university_coordinates

# We find some intersting finding in the dataset
# That is there are 748 Cementry citing but there are
# 1210 Haunted citing in School.
# Creepy
# Finding the location which have same latitude and longitude
# (without considering the NA values)

same_coordinates <- haunted_places %>%
  group_by(latitude, longitude) %>%
  filter(n() > 1) %>%
  ungroup()

# Remove NA values
same_coordinates <- same_coordinates %>%
  filter(!is.na(latitude) & !is.na(longitude))

same_coordinates

# Create a frequency table for the location column
same_coordinates_freq <- same_coordinates %>%
  count(location, sort = TRUE)

same_coordinates_freq
# Create a proportion table for the location column
same_coordinates_prop <- same_coordinates_freq %>%
  mutate(proportion = n / 870)

same_coordinates_prop


# We can see that some of the latitudes and longitudes are same for different
# location (but they are not, it is just the typo.)

# function to filter the duplicate data
filter_duplicate_locations <- function(df) {
  df %>%
    group_by(latitude, longitude) %>%
    filter(n_distinct(location) > 1) %>%
    arrange(latitude, longitude)         # Arrange by latitude and longitude
}

# Identify locations with the same latitude and longitude but different names
result <- filter_duplicate_locations(haunted_places)
```

```r
# Remove NA values
result <- result %>%
  filter(!is.na(latitude) & !is.na(longitude))

# Print the result
result

# Clean the location column
# Replace locations with the shortest name; if equal, use the first name
updated_location_names <- result %>%
  group_by(latitude, longitude) %>%
  mutate(location = location[which.min(nchar(location))]) %>%  # Shortest name
  ungroup()

# Print the updated data frame
updated_location_names
# Conforming it is working or not
updated_location_names <- filter_duplicate_locations(updated_location_names)

updated_location_names
nrow(haunted_places)
# Standardize location names only for non-NA coordinates
haunted_places <- haunted_places %>%
  group_by(latitude, longitude) %>%
  mutate(
    # Only standardize when coordinates are not NA
    standardized_location = if(!any(is.na(latitude)) &&
                               !any(is.na(longitude))) {
      names(which.max(table(location)))
    } else {
      location
    }
  ) %>%
  ungroup() %>%
  mutate(location = standardized_location) %>%
  select(-standardized_location)

# Verify the results
haunted_places %>%
  group_by(latitude, longitude) %>%
  filter(n_distinct(location) > 1,
         !is.na(latitude),
         !is.na(longitude))
# Frequency table for location column without NA values
location_freq <- haunted_places %>%
  filter(!is.na(location)) %>%
  count(location, sort = TRUE)


kable(location_freq[1:5,],
      caption = "Frequency and Proportional Table",
      col.names = c("Location", "Counts"))
# Group by latitude, longitude, and location, then count occurrences
same_coordinates_freq <- haunted_places %>%
  group_by(latitude, longitude, location) %>%
  summarise(count = n(), .groups = "drop") %>%  # drop grouping
  arrange(desc(count)) %>%  # Sort by count in descending order
  filter(!is.na(latitude) & !is.na(longitude))  # Remove NA values

# View the updated frequency table
same_coordinates_freq
top_50_places <- same_coordinates_freq[1:50,]
# Create the leaflet map of top 50 places with multiple haunted sightings
# Show a CUSTOM circle at each position. Size defined in Pixel.
# Size does not change when you zoom

m=leaflet(data = top_50_places) %>%
   addTiles() %>%
   addCircleMarkers(
      ~longitude, ~latitude,
      radius=~count*1 ,
      color=~ifelse(top_50_places$count>10 , "red", "orange"),
      stroke = TRUE,
      fillOpacity = 0.1,
      popup = ~as.character(location)
   )
m
# Converting the the leaflet map into image

# save it in html
saveWidget(m,"tmp_top_50.html",selfcontained = F)


# and in png or pdf
```

```r
webshot("tmp_top_50.html","top_50_map.png", delay =5, vwidth = 3840,
  vheight = 2160)

# save the haunted places data as a csv file
write_csv(haunted_places, "haunted_places.csv")
# test hypothesis
# Extract location type from 'location' column (if location types are embedded in text)
# Location Type Classification using reggex
haunted_places_hypo <- haunted_places %>%
  mutate(
    location_type = case_when(
      grepl("cemetery|graveyard|burial", location, ignore.case = TRUE) ~ "Cemetery",
      grepl("university|college|school|campus", location, ignore.case = TRUE) ~ "University"
      grepl("hospital|medical|clinic|asylum", location, ignore.case = TRUE) ~ "Hospital"
    )
  ) %>%
  # Remove NA values
  filter(!is.na(location_type), !is.na(state))


# contingency table
contingency_table <- table(haunted_places_hypo$location_type, haunted_places_hypo$state)

# Chi-Square Test of Independence
chi_test <- chisq.test(contingency_table)

# print the result
chi_test
# Visualize the contingency table
library(ggplot2)
contingency_df <- as.data.frame(contingency_table)
colnames(contingency_df) <- c("Location_Type", "State", "Count")

plot <- ggplot(contingency_df, aes(x = State, y = Count, fill = Location_Type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Distribution of Location Types by State",
       x = "State", y = "Count", fill = "Location Type") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

plot

ggsave("location_distribution_by_state.png", plot = plot, width = 10, height = 6, dpi = 300)
```

```r
# Function to calculate Haversine distance
haversine_distance <- function(lat1, lon1, lat2, lon2, radius = 6371) {
  # Convert degrees to radians
  to_radians <- function(deg) { deg * pi / 180 }

  lat1 <- to_radians(lat1)
  lon1 <- to_radians(lon1)
  lat2 <- to_radians(lat2)
  lon2 <- to_radians(lon2)

  # Haversine formula
  delta_lat <- lat2 - lat1
  delta_lon <- lon2 - lon1
  a <- sin(delta_lat / 2)^2 + cos(lat1) * cos(lat2) * sin(delta_lon / 2)^2
  c <- 2 * atan2(sqrt(a), sqrt(1 - a))

  # Distance in kilometers
  radius * c

}

# Remove 'description' column and filter out rows with NA values
haunted_places <- haunted_places %>%
  select(-description) %>%  # Remove the 'description' column
  drop_na()                 # Remove rows with NA values

haunted_places
# Calculate the distance and add as a new column
haunted_places <- haunted_places %>%
  rowwise() %>%
  mutate(
    distance_from_city = haversine_distance(
      latitude, longitude, city_latitude, city_longitude
    )
  ) %>%
  ungroup()
haunted_places %>% glimpse()
# Calculate distance from city center using Haversine formula
center_distance <- haunted_places %>%
  mutate(distance_from_center = haversine_distance(
    latitude, longitude,
    city_latitude, city_longitude
  ))

# Aggregate data by city
city_data <- center_distance %>%
  group_by(city) %>%
  summarize(
    total_haunted = n(),
    avg_distance = mean(distance_from_center, na.rm = TRUE)
  )

# Visualization
distance_plot <- ggplot(city_data, aes(x = avg_distance, y = total_haunted)) +
  geom_point(alpha = 0.7) +
  # geom_smooth(method = "lm", color = "blue", se = TRUE) +  # Added confidence interval
  labs(
    title = "Total Haunted Locations vs Distance from City Center",
    subtitle = "Using Haversine Distance Formula",
    x = "Average Distance from City Center (km)",
    y = "Total Number of Haunted Locations"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold"),
    plot.subtitle = element_text(color = "gray40")
  )
distance_plot

ggsave("distance_plot.png", plot = distance_plot, width = 10, height = 6, dpi = 300)

# First calculate the Haversine distances
center_distance <- haunted_places %>%
  mutate(distance_from_center = haversine_distance(
    latitude, longitude,
    city_latitude, city_longitude
  ))

# Create 2km bins and count haunted places in each bin
binned_data <- center_distance %>%
  mutate(
    # Create bins of 2km each
    distance_bin = cut(
      distance_from_center,
      breaks = seq(0, max(distance_from_center) + 2, by = 2),
      labels = seq(1, ceiling(max(distance_from_center)/2)) * 2 - 1  # Center points of bins
    )
  ) %>%
  group_by(distance_bin) %>%
  summarize(
    total_haunted = n(),
    avg_distance = as.numeric(as.character(distance_bin))  # Convert bin labels to numeric
  )


# Creating the model
# Linear regression model with binned data
bin_model <- lm(total_haunted ~ avg_distance, data = binned_data)

# Summary of the model
summary(bin_model)

# Visualization of binned data
ggplot(binned_data, aes(x = avg_distance, y = total_haunted)) +
  geom_point(alpha = 0.7, size = 3) +
  geom_smooth(method = "lm", color = "lightblue", se = TRUE) +
  labs(
    title = "Regression: Haunted Locations Count vs Distance from City Center",
    subtitle = "Using 2km Distance Bins",
    x = "Distance from City Center (km)",
    y = "Number of Haunted Locations"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold"),
    plot.subtitle = element_text(color = "gray40")
  )

# Diagnostic plots
png("diagnostic_plots.png", width = 800, height = 800)
par(mfrow = c(2, 2))
plot(bin_model)
```