Analyzing UFO Sightings

Shealagh Brown & Sam Zimpfer

2025-05-01

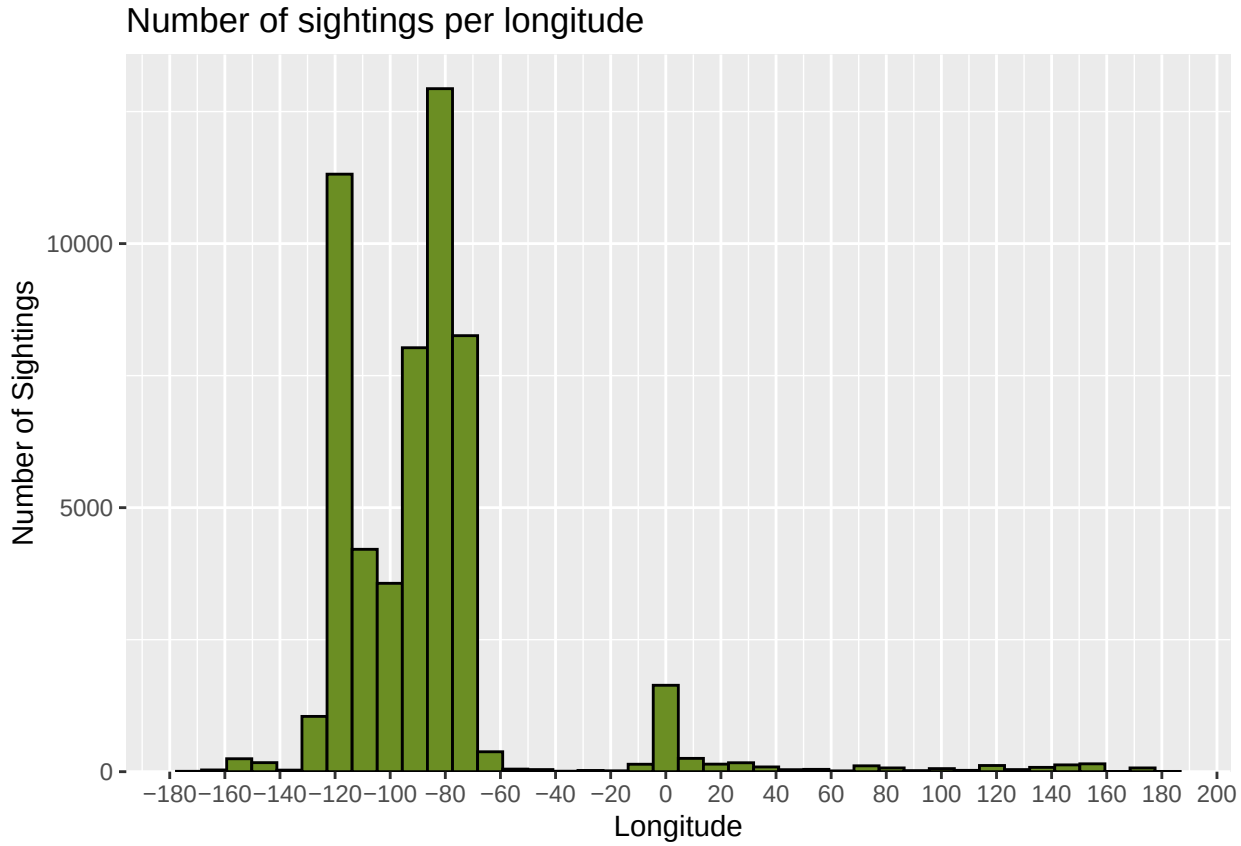
Introduction

The data we were working with came from a data set called "UFO sightings scrubbed" that was found on Kaggle.com from a user named Akhil Goyal. The data was last updated three months ago, making it quite recent data. It contains information on all UFO sightings dating back to 1906. The data could have some bias if ufo sightings from certain regions of the world were not recorded or included in this data set, additionally it is observational data collected by different people around the globe which can create large amounts of variation in the data.
This data is of interest because UFOs have been a topic of public debate for years. With increasing amounts of interest in space travel and extraterrestrials in more recent years, the fascinations with UFOs has only grown stronger. For this project we want to explore what influences sightings as this can be valuable knowledge for those trying to investigate UFOs.
In order to work with our data we had to clean it. This included converting the datetime column into year, month, day, seconds, minutes, hours format. Then we created a new data set where we added columns for years, seconds, and months and kept the updated datetime, city, state, country, longitude, and latitude columns. We then had to convert both longitude and latitude into numeric values in order to work with them. The code for this data cleaning follows:

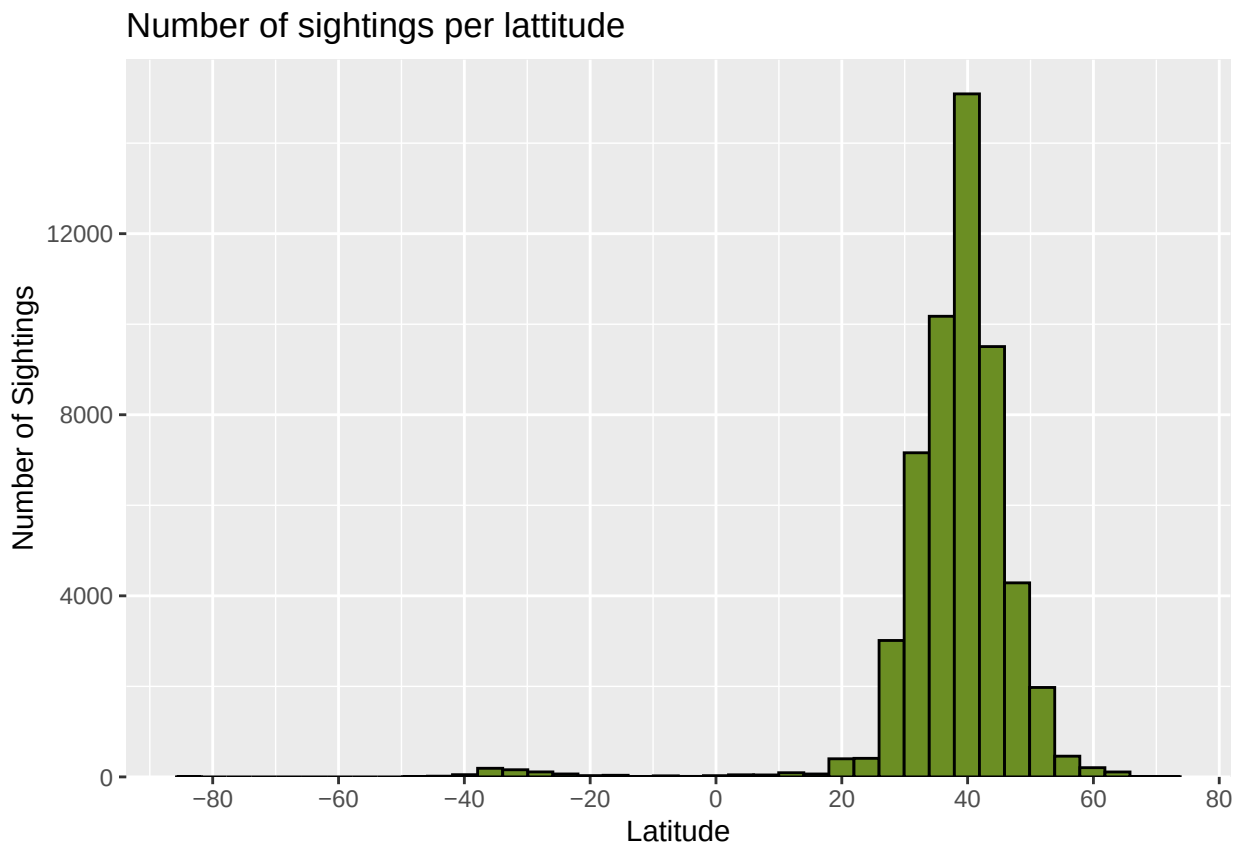
```
ufo_raw <- read.csv("ufo_sightings_scrubbed.csv")

#convert date time to ymd_hms format
ufo_raw$datetime <- ymd_hms(ufo_raw$datetime)

#cleaning data
ufo_raw|>
   mutate(seconds = duration..seconds.,
          year = year(datetime), #create year column
          month = month.name[month(datetime)])|> #create month column and convert to name
  select(datetime, city, state, country, seconds,latitude,longitude, year, month)|>
    filter(seconds <= 40000) |>
    # filter out badly formatted entries that could cause NA's during the following conversion
    filter(grepl("^[-]?[0-9.]+$", latitude),
           grepl("^[-]?[0-9.]+$", seconds)) -> ufo


ufo$latitude <- as.numeric(ufo$latitude) #changing lat to numeric
ufo$seconds <- as.numeric(ufo$seconds) #changing seconds to numeric

ufo <- ufo |> drop_na(longitude, latitude, seconds)
```

Data Analysis

The first question we had about UFO sightings is how the location around the world influenced the number of UFO sightings that occurred. We began by creating histograms for longitude and latitude. From there we created a scatter plot to compare latitude and longitude to see if there was a relationship between the two of them.



When looking at the histogram of longitude, we notice there are two main peaks within the spread. One peak is around -120 and the seconds is around -80. This make sense because these are the longitudes that correspond with highly populated areas of the US. The relative height of these peaks implies that there are considerably more sightings in the US than anywhere else in the world. This observation can be interpreted in multiple ways, either that UFO's are more commonly reported in the US than anywhere else (either accurately or inaccurately), or that there really are more UFO visits in the US than anywhere else. We can't make a solid determination between these two based on the data, but we can clearly see that there have been more reports around the US.



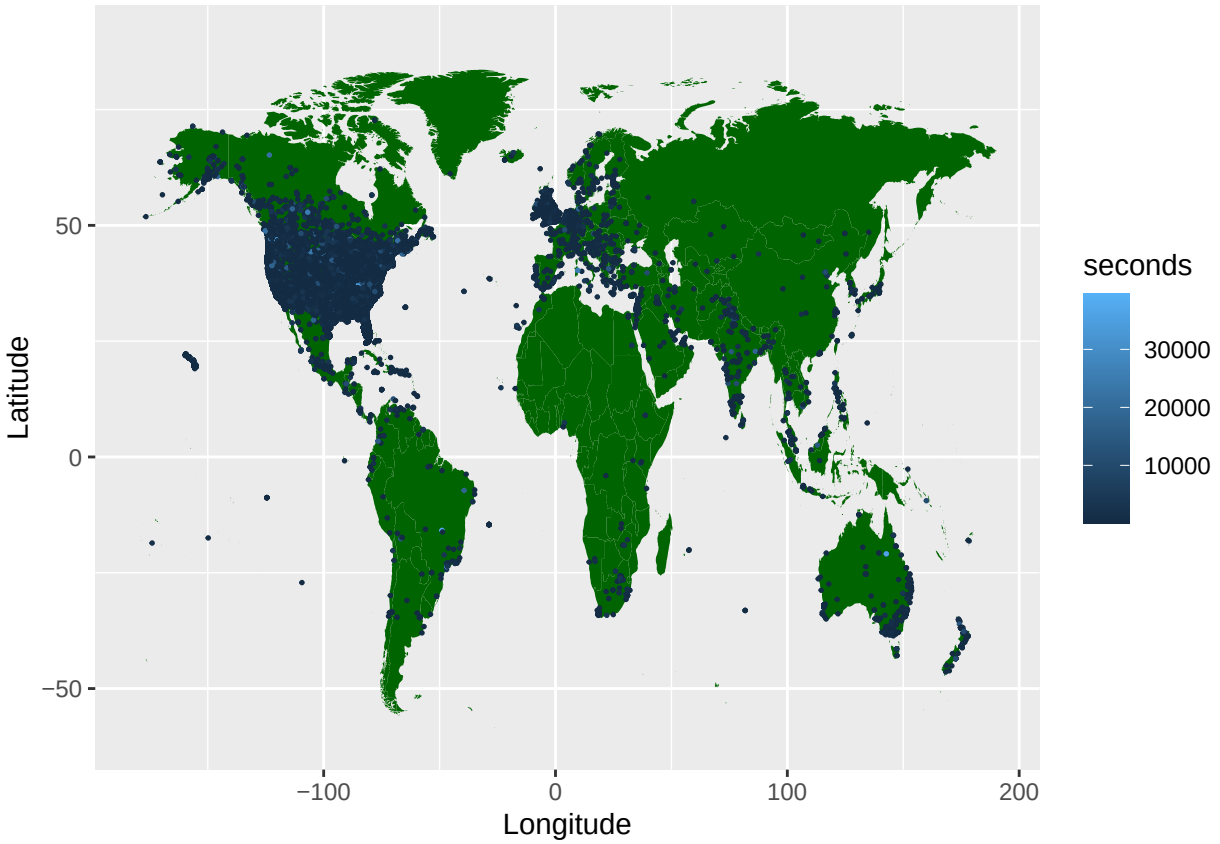
In the latitude histogram, we see than sightings are most prevalent around 40. Again, this corresponds to the coordinates of the US and also Europe, which is the second most frequently reported area of UFO sightings. This data supports the same conclusions we see from the longitude histogram.

```
map <- map_data("world")

map_plot <- ggplot()+
  geom_polygon(data = map,
               mapping = aes(x= long,
                             y = lat,
                             group = group),
               fill = "darkgreen")+
  geom_point(data = filter(ufo, seconds < 40000),
             mapping = aes(x = longitude,
                           y = latitude,
                           color = seconds),
             size = 0.3
           )+
  labs(
    x = "Longitude",
    y = "Latitude"
  )+
scale_y_continuous(expand = c(0, 0, 0.05, 0))+
  ylim(-60,90)
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```

```
map_plot
```

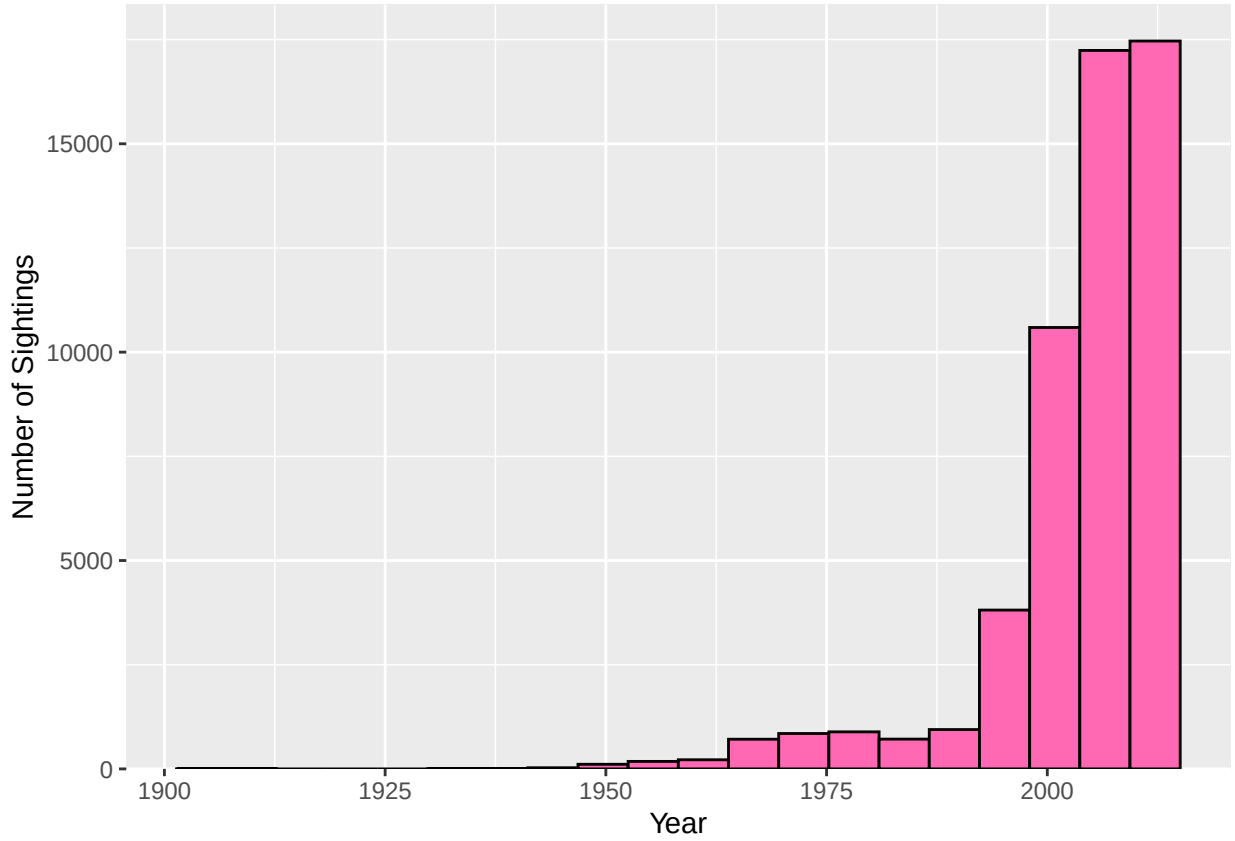
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



Here, we can see in greater detail the geographical dispersion of the sightings. The graph supports the same claims as the previous two graphs. We can also see from the color coding based on duration that most of the sightings are pretty short.

The next question we had was how the number of sightings had increase or decreased over the full epoch of time that the data set spans. We created a histogram of sightings per group of 5 years to answer this question.

```
ggplot( data = ufo,
        mapping = aes(x = year))+
  geom_histogram(fill = "hotpink",
                 color = "black",
                 bins = 20)+
  labs(
    x = "Year",
    y = "Number of Sightings"
  )+
scale_y_continuous(expand = c(0, 0, 0.05, 0))
```

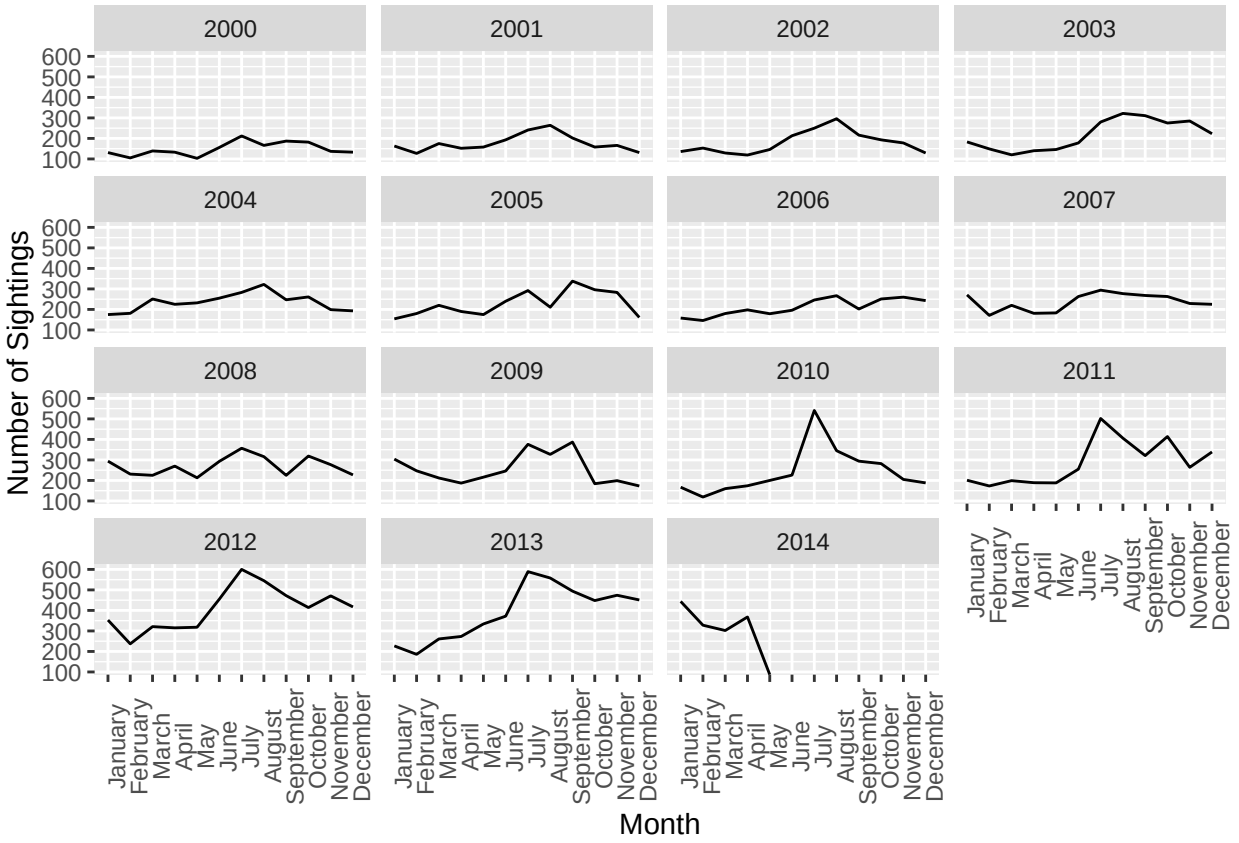


The histogram shows that UFO sightings were relatively infrequent in the early 1900's. Sightings gradually started to increase during the 1950's and then there was a sharp increase at the end of the 1990's and into the 2000's. The number of sightings seems to start to plateau past 2005 also, so it would interesting to see whether the data plateaus in the long term after this point or if it would continue to increase beyond a certain point. This pattern might indicate developments in human history that could be of interest to extraterrestrial beings or have sparked human interest in the existance of extraterrestrials.

```
# create table of year, month, number of sightings
summary <- ufo |> filter(year >= 2000) |> group_by(year, month) |> summarise(count = n())
```

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

```
# order data by natural month ordering
summary$month <- factor(summary$month, levels = month.name)
summary <- summary |> arrange(year, month)

ggplot(data = summary,
       mapping = aes(x = month,
                     y = count,
                     group = 1)) +
  geom_line() +
  labs(x = "Month",
       y = "Number of Sightings") +
  scale_y_continuous(expand = c(0, 0, 0.05, 0)) +
  theme(axis.text.x = element_text(angle = 90)) +
  facet_wrap(~ year)
```



Here we have a small multiples graph which shows the seasonal trends in UFO data since the year 2000. Like the previous graph, these graphs also show that total UFO sightings have increased since 2000. Before 2009 sightings are generally consistent over the years. In 2009 we can see the first real peak during the summer. This trend seems to be reflected in the following years since then. It's possible that this is simply because more people are outdoors at this time of year, so more UFO visits are reported as sightings. Because of this highly plausible explanation, we can't necessarily determine from the data whether there was an actual increase in UFO visits in the summers.

Machine learning

Our original goal was to use machine learning techniques to identify a relationship between the duration of sightings and the longitude/latitude the sightings took place at. We used both linear and kNN regression techniques to model this relationship, but after some trial and error it became apparent that the data points are spread in too random of a way to be modeled in any significant way by regression techniques. To see the absence of a linear relationship, we will take a look at the graphs produced below, and the code used to produce them. These graphs use a linear model based on only latitude, only longitude, and latitude plus longitude respectively.

```
lat_lm<- lm(
  formula = seconds ~ latitude,
  data = ufo
)

long_lm<- lm(
```

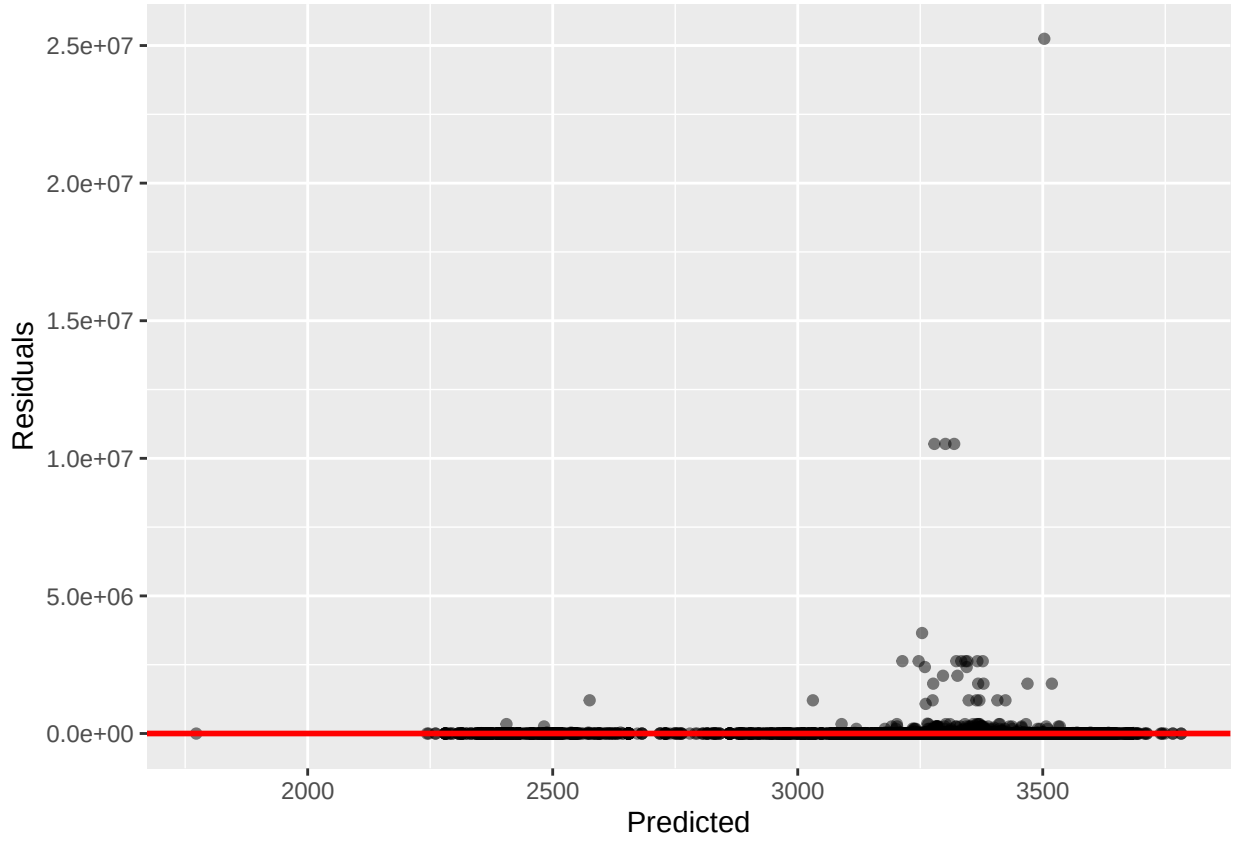
```
  formula = seconds ~ longitude,
  data = ufo
)

lat_long_lm<- lm(
  formula = seconds ~ latitude + longitude,
  data = ufo
)

bind_rows(
  glance(lat_lm),
  glance(long_lm),
  glance(lat_long_lm)
  ) |>

  mutate(
    explanatories = c(as.character(formula(lat_lm))[3],
                      as.character(formula(long_lm))[3],
                      as.character(formula(lat_long_lm))[3])
  ) |>
select(explanatories, r.squared, adj.r.squared, sigma)
```

```
## # A tibble: 3 x 4
##   explanatories          r.squared adj.r.squared   sigma
##   <chr>                      <dbl>         <dbl>   <dbl>
## 1 latitude             0.000000881    -0.0000177 142621.
## 2 longitude            0.0000784       0.0000598 142615.
## 3 latitude + longitude 0.0000999       0.0000627 142615.
```

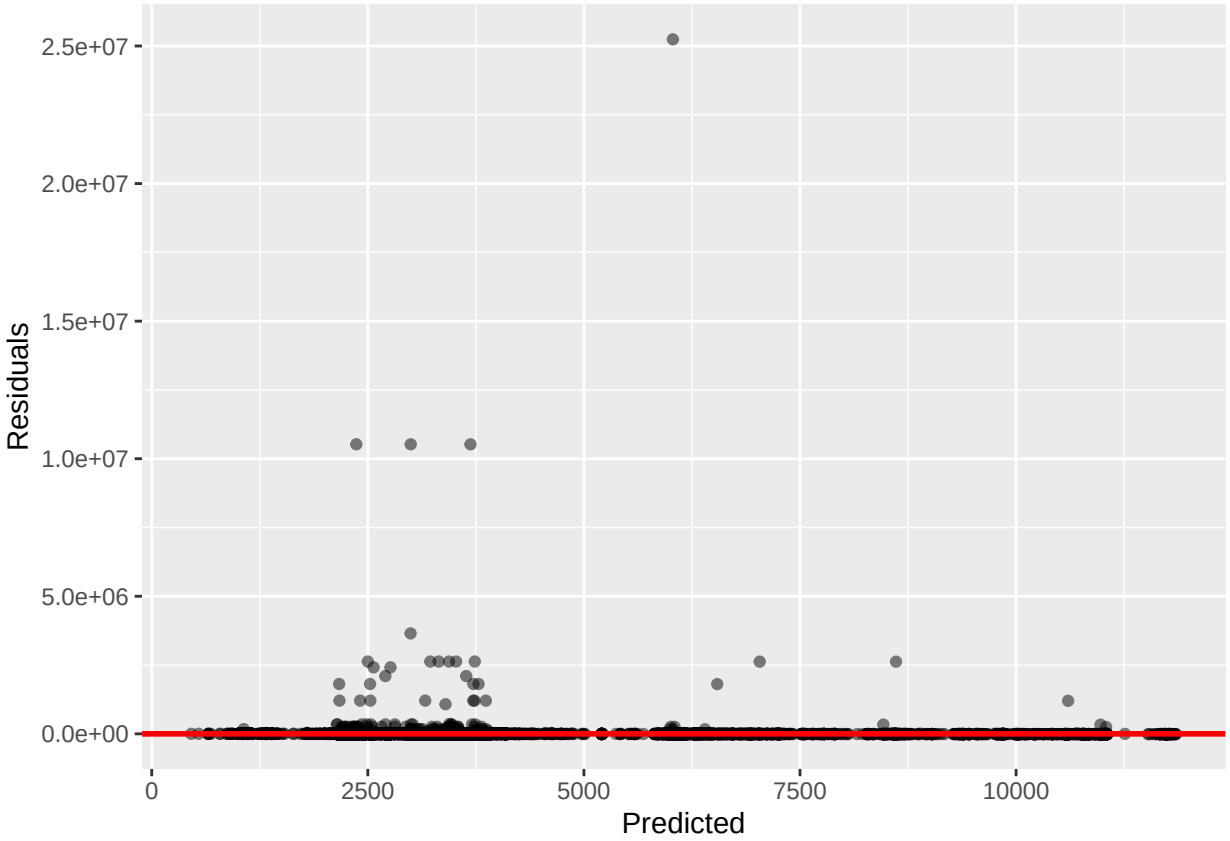
Residual Plots for Duration of Sightings vs Longitude and Lattitude

```
resid_plot_print <- function(model) {
  augment_columns(
  x = model,
  data = ufo
) |>

  ggplot(
    mapping = aes(
      x = .fitted,
      y = .resid
    )
  ) +

  geom_point(alpha = 0.5) +

  geom_hline(
    yintercept = 0,
    color = "red",
    linewidth = 1
  ) +
```

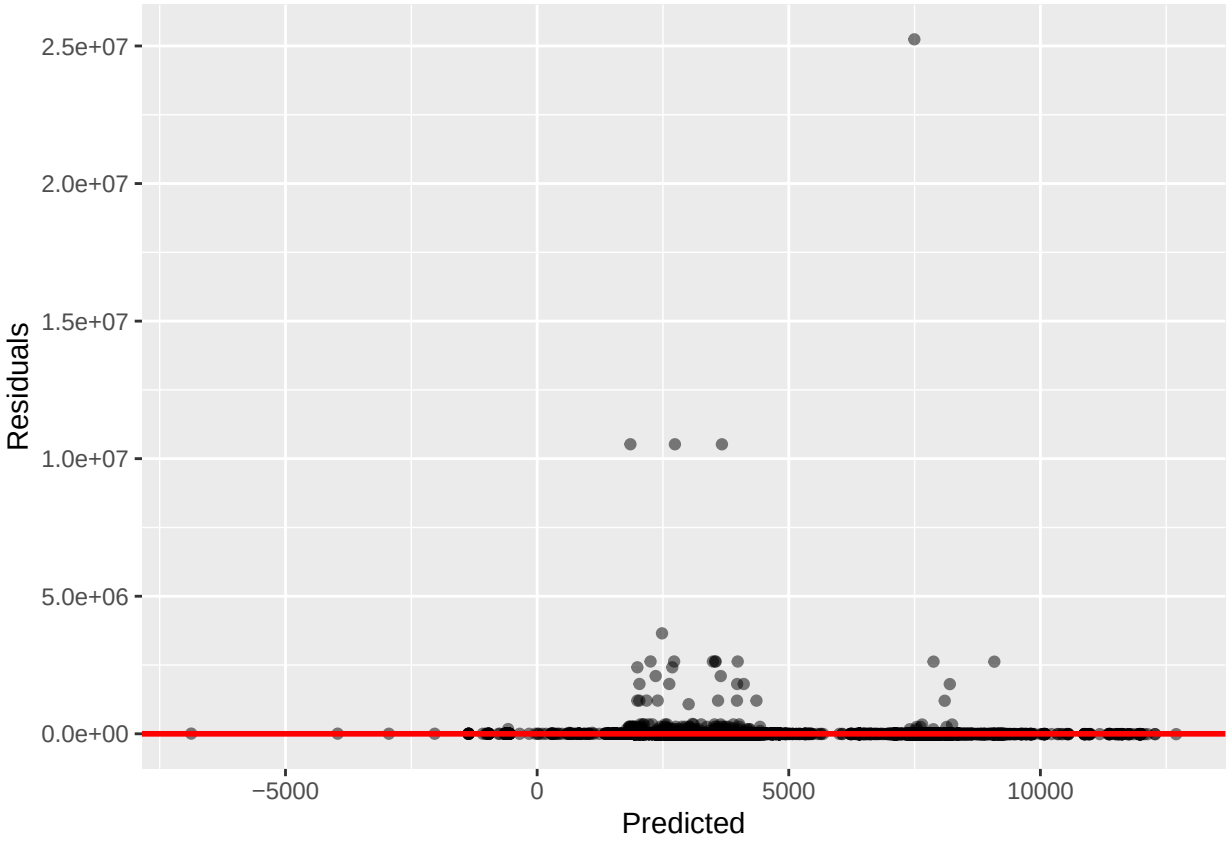
```
  labs(
    x = "Predicted",
    y = "Residuals"
  )}

resid_plot_print(lat_lm)
```




```
resid_plot_print(long_lm)
```




```
resid_plot_print(lat_long_lm)
```



In each graph, the mean of the residuals is very low, which is good because it means the average error in predictions is low, but there is a cluster of outlines in each graph that are severely under predicted. These clusters show that the data simply cannot be modeled linearly. To try and work around this, we attempted to use kNN regression instead. The process of identifying the best k value and normalization/standardization technique is shown below.

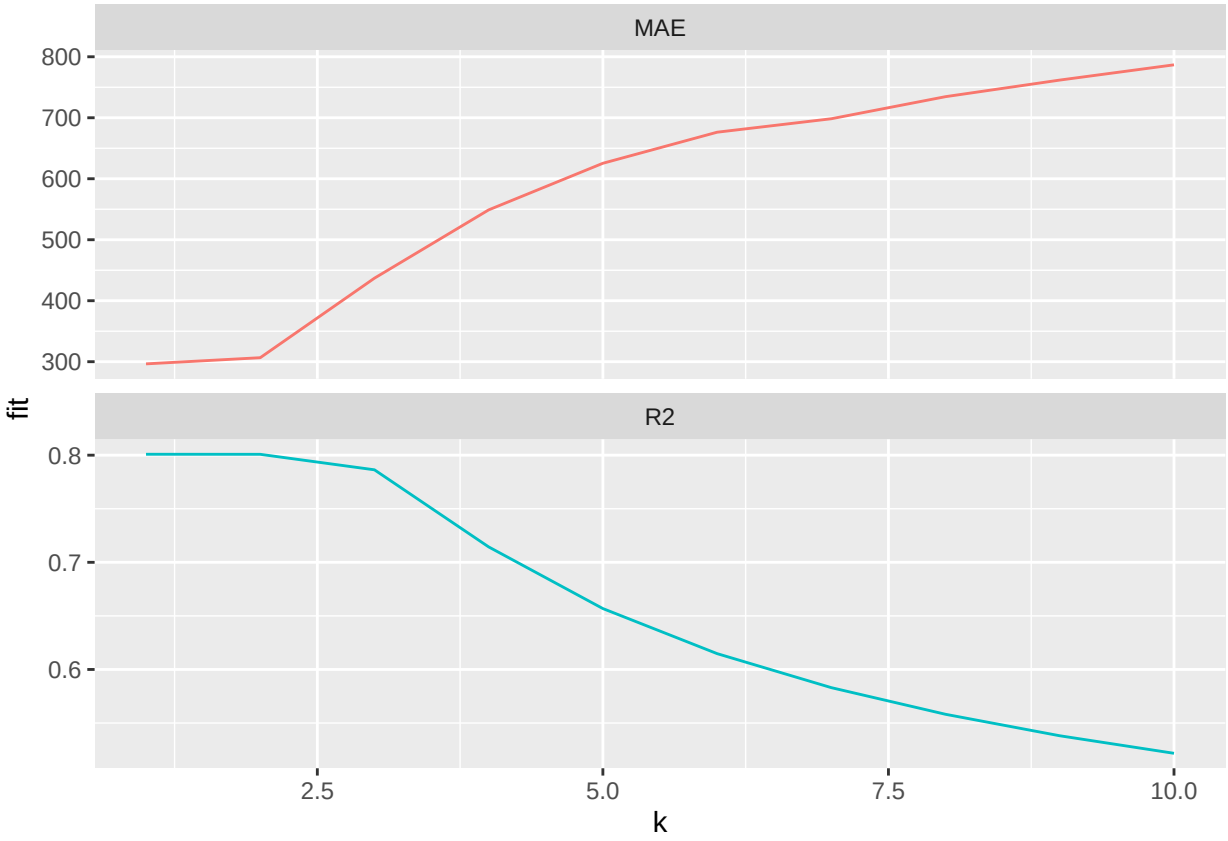
```
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}

standardize <- function(x) {
  return((x - mean(x)) / sd(x))
}

ufo_norm <-
  ufo |>
  select(longitude, latitude, seconds) |>
  mutate(
    across(
      .cols = latitude:longitude,
      .fns = normalize
    )
  )

ufo_stan <-
  ufo |>
```

```r
  select(longitude, latitude, seconds) |>
  mutate(
    across(
      .cols = latitude:longitude,
      .fns = standardize
    )
  )


k <- 1:10


# try with normalizing
fit_stats_norm <-
  tibble(k = k,
         R2 = rep(-1, length(k)),
         MAE = rep(-1, length(k)))

for (i in 1:length(k)) {
  norm_knn <-
    knn.reg(
      train = ufo_norm,
      y = ufo$seconds,
      k = k[i]
    )

  fit_stats_norm[i, "R2"] <- norm_knn$R2Pred
  fit_stats_norm[i, "MAE"] <- (ufo$seconds - norm_knn$pred) |> abs() |> mean()
}

fit_stats_norm
```

```
## # A tibble: 10 x 3
##        k    R2   MAE
##    <int> <dbl> <dbl>
##  1     1 0.801  296.
##  2     2 0.801  307.
##  3     3 0.786  437.
##  4     4 0.714  549.
##  5     5 0.657  625.
##  6     6 0.615  676.
##  7     7 0.583  698.
##  8     8 0.558  735.
##  9     9 0.538  762.
## 10    10 0.522  787.
```

```r
fit_stats_norm |>
  pivot_longer(
    cols = R2:MAE,
    names_to = "fit_stat",
    values_to = "fit"
  ) |>
```

```
  ggplot(mapping = aes(x = k,
                       y = fit,
                       color = fit_stat)) +
  geom_line(show.legend = F) +
  facet_wrap(facets = ~ fit_stat,
             scales = "free_y",
             ncol = 1)
```

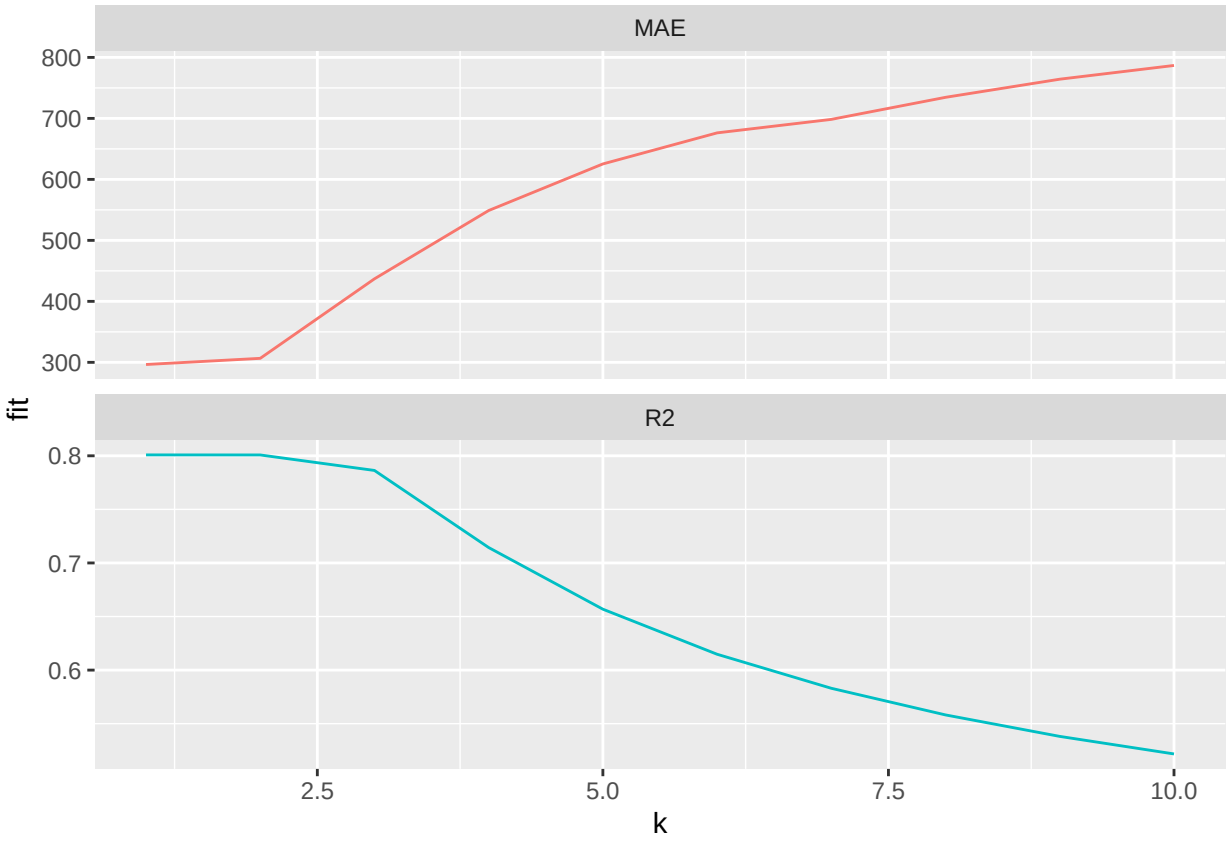



```
# try with standardizing
fit_stats_stan <-
  tibble(k = k,
         R2 = rep(-1, length(k)),
         MAE = rep(-1, length(k)))

for (i in 1:length(k)) {
  stan_knn <-
    knn.reg(
      train = ufo_stan,
      y = ufo$seconds,
      k = k[i]
    )

  fit_stats_stan[i, "R2"] <- stan_knn$R2Pred
  fit_stats_stan[i, "MAE"] <- (ufo$seconds - stan_knn$pred) |> abs() |> mean()
}
```

```
fit_stats_stan
```

```
## # A tibble: 10 x 3
##        k    R2   MAE
##    <int> <dbl> <dbl>
##  1     1 0.801  296.
##  2     2 0.801  307.
##  3     3 0.786  437.
##  4     4 0.714  549.
##  5     5 0.657  625.
##  6     6 0.615  676.
##  7     7 0.583  698.
##  8     8 0.558  735.
##  9     9 0.538  764.
## 10    10 0.522  787.
```

```
fit_stats_stan |>
  pivot_longer(
    cols = R2:MAE,
    names_to = "fit_stat",
    values_to = "fit"
  ) |>

  ggplot(mapping = aes(x = k,
                       y = fit,
                       color = fit_stat)) +
  geom_line(show.legend = F) +
  facet_wrap(facets = ~ fit_stat,
             scales = "free_y",
             ncol = 1)
```




```
fit_stats_combined <-
  bind_rows("stan" = fit_stats_stan,
            "norm" = fit_stats_norm,
            .id = "rescale")

fit_stats_combined |>
  filter(
    R2 == max(R2) | MAE == min(MAE)

  )
```

```
## # A tibble: 1 x 4
##   rescale     k    R2   MAE
##   <chr>   <int> <dbl> <dbl>
## 1 norm        1 0.801  296.
```

```
norm_knn <-
  knn.reg(
    train = ufo_norm,
    y = ufo$seconds,
    k = 1
  )
```

The fifth table show that the best k value to use is 1, which is typically not a value that produces accurate regression models. Because of this, we see that kNN regression is also not a viable method of machine learning.