Final Project - Implementation Phase

Title of Final Project: 2017 Mini-Challenge 1

Team Name: The Visualizers (Section 1, 1pm-2pm)

Name: Misty Peng

UT EID: mp46528

Name: Brittany Given

UT EID: btg587

Date: 4/5/2023

```
import pandas as pd
import numpy as np
# altair
import altair as alt
# plotly
import plotly.graph_objects as go
import plotly.express as px
```

Project Outline

Overview

Given a dataset on gate sensor data of Lekagul Nature Preserve, which tracked vehicles as they moved through gates in the preserve. The types of gates tracked include campsites, gates restricted to park rangers, and entrances. The car ID, car type, and timestamp were all tracked as well. Data was tracked for about a year between 2015 and 2016.

Goal of the Project

Our goal is to try and determine what could be most impactful to bird life in the nature preserve.

Notes about the Preserve

- Speed limit is 25 mph, with the shortest route entrance to entrance being about 12 miles
- Preserve is open 24 hours a day year long
- Certain sections are restricted to certain car types

Our Plan of Action

- Look at travel patterns
- Determine irregularities to these patterns
- Observed travel patterns that would have consequences to bird life

Summary of Our Overall Findings

Overall, the top three potential impacts on the species in the park are:

1. High periods of activity in the park resulting in pollution/disturbances
2. Pollution from large vehicles venturing in restricted areas
3. Disturbances from speeding vehicles

Analysis

Description of the Dataset

```
# read in data file
df = pd.read_csv("Lekagul Sensor Data.csv")

# get size
print(df.shape)

# get basic information about df, such as head/tail
print(df.head(5))
```

```
df.tail(5)
```

```
(171477, 4)
             Timestamp              car-id car-type      gate-name
0  2015-05-01 00:43:28  20154301124328-262        4      entrance3
1  2015-05-01 01:03:48  20154301124328-262        4  general-gate1
2  2015-05-01 01:06:24  20154301124328-262        4   ranger-stop2
3  2015-05-01 01:09:25  20154301124328-262        4   ranger-stop0
4  2015-05-01 01:12:36  20154301124328-262        4  general-gate2
```

	Timestamp	car-id	car-type	gate-name
171472	2016-05-31 23:40:13	20161031111001-854	6	ranger-stop2
171473	2016-05-31 23:42:08	20165831105856-579	1	general-gate4
171474	2016-05-31 23:43:13	20161031111001-854	6	general-gate1
171475	2016-05-31 23:49:45	20165831105856-579	1	general-gate7
171476	2016-05-31 23:56:06	20161031111001-854	6	general-gate4

Manipulation of the Dataset

```
# add a new column to the dataset to show gate-type, since each gate type has known vehicles which should be traveling through it

# define list to hold data of new column
gateType = []
# loop through rows to gather data for each row
for idx, row in df.iterrows():
  # get gate-type information
  gt = row['gate-name'][0:-1]
  gateType.append(gt)
# add new column to the dataframe
#print(gateType[0:10])
df['gate-type'] = gateType

df.head(10)
```

	Timestamp	car-id	car-type	gate-name	gate-type
0	2015-05-01 00:43:28	20154301124328-262	4	entrance3	entrance
1	2015-05-01 01:03:48	20154301124328-262	4	general-gate1	general-gate
2	2015-05-01 01:06:24	20154301124328-262	4	ranger-stop2	ranger-stop
3	2015-05-01 01:09:25	20154301124328-262	4	ranger-stop0	ranger-stop
4	2015-05-01 01:12:36	20154301124328-262	4	general-gate2	general-gate
5	2015-05-01 01:24:02	20154301124328-262	4	general-gate5	general-gate
6	2015-05-01 01:31:41	20153101013141-937	1	entrance3	entrance
7	2015-05-01 01:33:57	20154301124328-262	4	entrance4	entrance
8	2015-05-01 01:53:34	20153101013141-937	1	general-gate1	general-gate
9	2015-05-01 01:56:20	20153101013141-937	1	ranger-stop2	ranger-stop

Find the traffic patterns associated with each gate based on the time of year

Questions we are curious about:

- Are there patterns in the times vehicles typically travel to the park?
- Are there particular gates that are more popular in its gate-type grouping than others? Is there a reason for this or pattern to it?

```
# analyze "patterns of life" —> view how visits to the park are spread out over time and over each gate

# allow all rows to show
alt.data_transformers.enable('default', max_rows=None)

# define interation for graph
click = alt.selection_multi(encodings=['y'])

# get values for car-types for bar chart
carTypes=list(sorted(set(df['car-type'].to_list())))
#print(carTypes)

# create scatter plot
scatter = alt.Chart(df, title='Sensor Readings over Time per Gate').mark_tick().encode(
```

```
    alt.Y('gate-name:N',
          axis=alt.Axis(title="Gate Name")),
    x='Timestamp:T',
    color=alt.condition(
        click,
        alt.Color('gate-type:N'),
        alt.value('lightgray'))
).properties(
    width=900
).add_selection(
    click
)

# create line chart to show total count over time
line = alt.Chart(df, title='Sensor Readings per Day based on Selected Gate').mark_line(point=True).encode(
    alt.X('yearmonthdate(Timestamp):T',
          axis=alt.Axis(title="Timestamp")),
    y='count()', # uncomment if we don't want y axis to have set max limit
    # alt.Y('count()',
    #       scale=alt.Scale(domain=(0,1800))),
    tooltip=[alt.Tooltip('yearmonthdate(Timestamp):T',title='Timestamp'), 'count()']
).properties(
    width=900
).transform_filter(
    click
)

# vertically concatenate
alt.hconcat(scatter, line)
```

Analysis of above Figures

- Figure 1 -

The first figure ("Sensor Readings over Time per Gate"), shows the overall use of the gates, entrances, and camping areas for all vehicles entering, traveling through, and staying overnight in the Preserve over time. We want to analyze the common patterns across the gates.

Observations Right off the bat, this figure shows us that there is a seasonal increase during the summer months, when it is warmer and outside activity is more popular, as compared to the winter months. This is seen in the patterns associated with gates of type "camping" and "entrance".

- Figure 2 -

To better view the frequency of vehicles over each day tracked, the second figure ("Sensor Readings per Day") shows a linechart of the count of vehicles moving through any of the gates in the park per day.

Observations Here, we can also see that, overall, there is high activity in the park over the summer months.

Interactivity has been included in the graph to allow the user to select a gate in the first figure and then have the line chart filter the data relating to only that gate-name's sensor. Thus, we can observe if there are any trends in the amount of cars going through a specific gate over time.

Overall Observations of Patterns & Irregularities

Expectations There is no difference between the camping, entrance, and general-gate gate-types. Patterns should be similar across the different gate-types. If not, what is the reasoning for the differences? For instance, is it the location of the sites or popularity or other?

Irregularities

- "camping1" is not used much at all, especially compared to patterns found on all other camping sites
- "general-gate0", "general-gate3", and "general-gate6" are not used as much compared to other gates in this gate-type

Find the visitation patterns associated with the day of the week and time of day

Questions we are curious about:

- Are there patterns in the time of day vehicles typically travel to the park?
- Are there patterns in the day of week vehicles typically travel to the park?

```
# are certain hours of the day more popular to visit?

# allow all rows to show
alt.data_transformers.enable('default', max_rows=None)

df['Timestamp'] = pd.to_datetime(df['Timestamp'])
df['month'] = df['Timestamp'].dt.month_name()

# dropdown filter for the month
months = [None, 'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
month_dropdown = alt.binding_select(options=months, name="Select Month: ")
```

```
month_select = alt.selection_single(fields=['month'], bind=month_dropdown)

# create the heatmap
alt.Chart(df, title='Sensor Activity Heatmap by Month').mark_rect().encode(
    alt.X('hours(Timestamp):T', title='Hour of the Day'),
    alt.Y('day(Timestamp):N', title='Day of the Week'),
    alt.Color('count():Q', title='Number of Sensor Readings'),
    tooltip=['count():Q']
).add_selection(
    month_select
).transform_filter(
    month_select
).properties(
    width = 800,
    height = 400
)
```

Analysis of above Figures

- Figure 1 -

The figure ("Sensor Activity Heatmap by Month"), shows the breakdown of activity within the park by day of the week and hour of the day.

Observations This figure shows us which times of the week are most popular for activity within the park. Additionally, we can see how these trends change from month to month.

Interactivity has been included in the graph to allow the user to select a month to limit the heatmap to only activity within that month.

Overall Observations of Patterns & Irregularities

Expectations The park is open 24 hours a day and year round. It is popular for professional and amateur photographers, students, and the general public. We would think that there would be more activity on the weekends, just because there would be more freetime for children and parents to visit the park. There could be high activity during the weekday that could be attributed to students going on field trips (or the like) in the park.

Irregularities We are surprised by the general high frequency of activity during the weekdays in the park. This varies from month to month. For instance...

- The month of September has a high amount of activity during all days of the week, whereas with a month like November, we do see our expected pattern of greater activity to be limited to the weekend (like Saturdays and Sundays)
- In the month of July, we see the highest number of sensor readings on Fridays, leading us to believe that there is a recurring activity on Fridays during July.

*****

POTENTIAL IMPACT #1

High periods of activity within the park causing pollution/disruption: *There does to appear to be regular events occuring during specific times/days during certain months or just sudden influxes of people during certain hours of the day on certain days of the week in certain months. Human activity is known to be a burden to species' habitats and is likely one of the culprits to the Nature Preserve's species.*

*****

Find the car-types that visit each gate to see if there are irregularities

Questions we are curious about:

- Do we see any irregularities, such as car-types visiting gates that they should not be at?

```
# analyze "car visit patterns to gates" -> view which types of cars visit which gates

# allow all rows to show
alt.data_transformers.enable('default', max_rows=None)

# define interation for graph
click = alt.selection_multi(encodings=['x'])

# get values for car-types for bar chart
carTypes=list(sorted(set(df['car-type'].to_list())))
#print(carTypes)

# create bar chart of the total visits to each gate name
bar1 = alt.Chart(df, title='Total Sensor Readings per Gate').mark_bar().encode(
    alt.X('gate-name:N',
```

```
          axis=alt.Axis(title="Gate Name")),
    y='count()',
    color=alt.condition(
        click,
        alt.Color('gate-type:N'),
        alt.value('lightgray')),
    tooltip=['count()']
).properties(
    height=500
).add_selection(
    click
)

# create bar chart of sensor readers per car type per selected gate
bar2 = alt.Chart(df, title='Total Sensor Readings per Car Type based on Selected Gate').mark_bar().encode(
    alt.X('car-type:N',
          scale=alt.Scale(domain=carTypes),
          axis=alt.Axis(title="Car Type")),
    y='count()',
    # alt.Y('count()',
    #       scale=alt.Scale(domain=(0,70000))),
    color=alt.value('lightblue'),
    tooltip=['count()']
).properties(
    height=500
).transform_filter(
    click
)

# vertically concatenate
alt.hconcat(bar1, bar2)
```

Analysis of above Figures

- Figure 1 -

The first figure ("Total Sensor Readings per Gate"), shows the overall use of the gates, entrances, and camping areas for all vehicles entering, traveling through, and staying overnight in the Preserve. We want to analyze the pattern of car-types to visit each gate, as each gate-type is utilized a particular way and has restrictions around the type of cars that should be passing through the gate. This figure shows us which gates in each gate-type grouping have the most activity.

- Figure 2 -

The second figure ("Total Sensor Readings per Car Type based on Selected Gate") shows the frequency of sensor readings per car type in the park.

Interactivity has been included in the graph to allow the user to select a gate in the first figure and then have the bar chart to the right to filter the data relating to only that gate-name's sensor. Thus, we can observe the frequency of car types at each gate and if there are any car-types traveling through gates they should not be at.

Overall Observations of Patterns & Irregularities

Expectations All vehicles are allowed to pass through gate-types "entrance", "general-gate", and "camping". Only Preserve Ranger vehicles should be passing through gate-types "gate" and "ranger-stop", which is car-type "2P". However, due to the placement of some ranger-stops, there is possibility of some ranger-stops recording all passing traffic. These include ranger-stop0, ranger-stop2, and ranger-stop4.

Other expectations we have include:

- "ranger-stop1" is limited to the public by "gate2", so if someone is going through "ranger-stop1", they should also be going through "gate2"
- "ranger-stop3" is limited to the public by "gate3", so if someone is going through "ranger-stop3", they should also be going through "gate3"
- "ranger-stop6" is limited to the public by "gate5" and "gate6", so if someone is going through "ranger-stop6", they should also be going through "gate5" or "gate6"

Irregularities

- "gate3" has recordings from car-type "4", which is a 4 axle (and above) truck, in addition to "2P"
- "gate5" has recordings from car-type "4", which is a 4 axle (and above) truck, in addition to "2P"
- "gate6" has recordings from car-type "4", which is a 4 axle (and above) truck, in addition to "2P"
- "ranger-stop1" has recordings from car-type "1", which is a 2 axle car (or motorcycle), in addition to "2P"
- "ranger-stop3" has recordings from car-type "4", which is a 4 axle (and above) truck, in addition to "2P"
- "ranger-stop6" has recordings from car-type "4", which is a 4 axle (and above) truck, in addition to "2P"
- "ranger-stop1" has recordings but there are no recordings for passage of prohibited vehicles through "gate2"

*****

POTENTIAL IMPACT #2

Faulty gates preventing traffic are not working: *Since "gate2" does not have sensor readings for prohibited vehicles, but we see cars are getting to "ranger-stop1", which requires one to technically go through "gate2", a potential impact to these birds are that restricted areas are not being well guarded from visitors. Perhaps some of the sensors are not reading cars correctly, from being broken or turned off, or the gates there does not actually exist like the map shows.*

*****

Analyze irregularities in gate-types "gate" and "ranger-stop" further

Questions we are curious about:

- Are there certain repeat offenders that are visiting gates and ranger-stops when they should not be?
- Is there a potential pattern in the wrong vehicles visiting gates and ranger-stops?

```
# view gate3, gate5, gate6, ranger-stop1, ranger-stop3, and ranger-stop6
# we want to see if there are certain vehicles that keep traveling to these gates when they should not be

# filter the dataframe to only show the above gate-names
gateNameList = ['gate3', 'gate5', 'gate6', 'ranger-stop1', 'ranger-stop3', 'ranger-stop6']
df_gateFilter = df[(df['gate-name'].isin(gateNameList)) & (df['car-type']!='2P')]
df_gateFilter.head(10)

# allow all rows to show
alt.data_transformers.enable('default', max_rows=None)

# define interation for graph
click = alt.selection_multi(encodings=['y'])

# create bar chart
bar1 = alt.Chart(df_gateFilter, title='Total Sensor Readings per Gate').mark_bar().encode(
    alt.Y('gate-name:N',
          axis=alt.Axis(title="Gate Name")),
    x = 'count()',
    color=alt.condition(
        click,
        alt.value('lightblue'),
        alt.value('lightgray')),
    tooltip=['count()']
).add_selection(
    click
).properties(
    width=900
)

# get values for car-types for bar chart
carTypes=list(sorted(set(df['car-type'].to_list())))
#print(carTypes)

# barplot that shows the car-id and number of times it visited a gate
bar2 = alt.Chart(df_gateFilter, title='Visits by Car based on Gate Selection').mark_bar().encode(
    alt.Y('car-id:N',
          axis=alt.Axis(title="Car ID")),
    alt.X('count()', scale=alt.Scale(domain=[0,10])),
    color=alt.Color('car-type:N',
          scale=alt.Scale(
              domain=['1','4'],
              range=['orange', 'pink']
              )),
    tooltip=[alt.Tooltip('car-type:N',title='Car Type'),
             alt.Tooltip('car-id:N',title='Car ID')]
).transform_filter(
    click
).properties(
    width=500
)

alt.vconcat(bar1, bar2)

# ck = df_gateFilter[(df_gateFilter['gate-name']=='ranger-stop6')]
# ck.shape
```

*****

POTENTIAL IMPACT #2 CONTINUED

The above table shows us that motorcycles (car-type == 1) are the only ones visiting "ranger-stop1". Perhaps the gate sensor is not registering visits from motorcycles or motorcycles are finding a way to get around the gate sensor reading.

*****

```
df_gateFilter.head(10)
```

Analyze irregularities in gate-types "gate" and "ranger-stop" further, specifically looking at the timeline of visits

Questions we are curious about:

- Is there a pattern that these offenders are visiting prohibited gates?

```
# view timeline of visits to gate3, gate5, gate6, ranger-stop1, ranger-stop3, and ranger-stop6

cars = alt.Chart(df_gateFilter, title='Total Sensor Readings by Car to Prohibited Gates').mark_bar().encode(
    alt.Y('car-id:N',
          axis=alt.Axis(title="Car ID")),
    x = 'count()',
    color=alt.condition(
        click,
        alt.Color('car-type:N'),
        alt.value('lightgray')),
    tooltip=['count()']
).add_selection(
    click
).properties(
    width=900
)

# set the range of the timestamp for x-axis
timestampDomain = [min(df_gateFilter['Timestamp']), max(df_gateFilter['Timestamp'])]
gates = alt.Chart(df_gateFilter, title='Timeline of Visits to Prohibited Gates Based on Car Selection').mark_circle().encode(
    alt.X('yearmonthdatehoursminutesseconds(Timestamp):T',
          axis=alt.Axis(title="Timestamp")),
    #alt.X('yearmonthdate(Timestamp):T', scale=alt.Scale(domain=timestampDomain)),
    alt.Y('gate-name', scale=alt.Scale(domain=gateNameList),
          axis=alt.Axis(title="Gate Name")),
    tooltip=[alt.Tooltip('car-id:N',title='Car ID'),
             alt.Tooltip('yearmonthdatehoursminutesseconds(Timestamp):T', title='Timestamp'),
             alt.Tooltip('gate-name:N',title='Gate Name')]
).transform_filter(
    click
).properties(
    width=900
)

alt.vconcat(cars, gates)
```

Analysis of above Figures

- Figure 1 -

The first figure ("Total Sensor Readings by Car to Prohibited Gates"), shows the frequency of sensor readings per car type in the park. The information in this chart has been filtered to only show cars visiting prohibited gates.

- Figure 2 -

The second figure ("Timeline of Visits to Prohibited Gates Based on Car Selection") shows the timeline of the selected car visiting the prohibited gates.

Interactivity has been included in the graph to allow the user to select a car in the first figure and then have the bar chart to the bottom filter the data relating to only sensors relating to that car-id. Thus, we can observe travel pattern that these cars are taking in relation to the gates they should not be at.

Overall Observations of Patterns & Irregularities

- All vehicles of "car-type" 1, which are motorcycles enter and exit the ranger-stop1.
- All vehicles of "car-type" 4, which are 4 axle (and above) trucks, move from "gate6" -> "ranger-stop6" -> "gate5" -> "gate3" -> "ranger-stop3" -> "ranger-stop3" -> "gate3" -> "gate5" -> "ranger-stop6" -> "gate6"

*****

POTENTIAL IMPACT #3

Pollution from large vehicles: *Large cars 4 axles (and above) trucks are not casually drive vehicles. These are all commercially large vehicles or a car with a trailer. Considering cars with trailers would likely be camping (and thus heading to a camp site), we believe that the vehicles that are passing through this route are commercial trucks, perhaps with certain necessary deliveries to the ranger-stops. However, these large vehicles would cause a great deal of distress to the protected areas they are passing through, in terms of all weight shifting vegitation, deisel/gasoline harming the air, and sound.*

*****

Find vehicles traverse through the preserve the most

Questions we are curious about:

- How many cars have frequent visits?
- Do the frequent visitors follow a pattern of where they go in the park?

```
from pandas.io.parsers.python_parser import count_empty_vals

# import collections for counter
import collections

# remove cars of type '2P' from the df
df_noRanger = df[(df['car-type']!='2P')]
#print(df_noRanger.head(5))

# check only sensors when people are coming in/out of the park
df_enterExit = df_noRanger[(df_noRanger['gate-type']=='entrance')]

# get counts of how often people visit
#cars = df_noRanger['car-id'].tolist()
cars = df_enterExit['car-id'].tolist()
carsCount = collections.Counter(cars)
#print(carsCount)

# remove the cars that only visit once (meaning d.value = 2)
freqVisitors = {k:int(v/2) for k,v in carsCount.items() if v > 2}
#print(freqVisitors)

# get info for freq visitors
df_freqVisitors = df[(df['car-id'].isin(freqVisitors.keys()))]
#print(df_freqVisitors.head())

# # add the frequency count to df to make easier to make interactive histogram
# # define list to hold data of new column
# count = []
# # loop through rows to gather data for each row
# for idx, row in df.iterrows():
#   # get car-id
#   cid = row['car-id']
#   # get gate-type information
#   count.append(carsCount[cid])
# # add new column to the dataframe
# #print(gateType[0:10])
# df['total-visits'] = count
# #print(df.head(10))

# get the car-type of the car-ids
carType = []
for k in freqVisitors.keys():
  row = df_enterExit[(df_enterExit['car-id']==k)].iloc[0]
  carType.append(row['car-type'])

# create new dataframe of information from collections
#d = {'car-id': carsCount.keys(), 'count': carsCount.values()}
d = {'car-id': freqVisitors.keys(), 'car-type': carType, 'count': freqVisitors.values()}
df_fv = pd.DataFrame.from_dict(d)
# sort by count
#df_fv = df_fv.sort_values(by=['count'], ascending=False)
#print(df_fv.head())

# allow all rows to show
alt.data_transformers.enable('default', max_rows=None)

# define interation for graph
click = alt.selection_multi(encodings=['y'])

# create bar chart showing car-ids with frequent visits
bar1 = alt.Chart(df_fv, title='Visits to Park by Car ID for Frequent Visitors').mark_bar().encode(
    alt.X('count:Q',
```

```
          axis=alt.Axis(
              values=list(range(
                  0,
                  max(df_fv['count'].to_list())+2,
                  2)
              ))),
    y = 'car-id:N',
    color=alt.condition(
        click,
        alt.Color('car-type:N'),
        alt.value('lightgray')),
    tooltip=['count:Q']
).add_selection(
    click
)

# create bar chart showing which gates the cars with frequent visits are traveling to
bar2 = alt.Chart(df_freqVisitors, title='Sensor Readings per Gate based on Selected Car ID').mark_bar().encode(
    x='count()',
    y='gate-name:N',
    color=alt.value('lightblue'),
    tooltip=['count()']
).transform_filter(
    click
)

alt.hconcat(bar1, bar2)
```

Analysis of above Figures

- Figure 1 -

The first figure ("Visits to Park by Car ID for Frequent Visitors"), shows the number of times a frequent visitor visited the park, with a "frequent visitor" being someone who visits the park more than once during the data collection time period.

Observations This figure shows us that not many cars visit the park more than once often. However, for those that do, there are outliers of people that apparently love to visit the park often. One car visits the park 16 times. The next most frequent visitor visits 7 times.

- Figure 2 -

The second figure ("Sensor Readings per Gate based on Selected Car ID") shows the frequency of sensor readings per gate in the park, only for the most frequent visitors.

Interactivity has been included in the graph to allow the user to select a car in the first figure and then have the bar chart to the right filter the data relating to only that car-id. Thus, we can observe where these frequent visitors are going and if they are going to places they should not be at.

Overall Observations of Patterns & Irregularities

Expectations All vehicles are allowed to pass through gate-types "entrance", "general-gate", and "camping". Only Preserve Ranger vehicles should be passing through gate-types "gate" and "ranger-stop", which is car-type "2P". However, due to the placement of some ranger-stops, there is possibility of some ranger-stops recording all passing traffic. These include ranger-stop0, ranger-stop2, and ranger-stop4.

Irregularities We do not find a lot of irregularity viewing the visits of frequent visitors.

View Length of Time Cars Stay in the Preserve

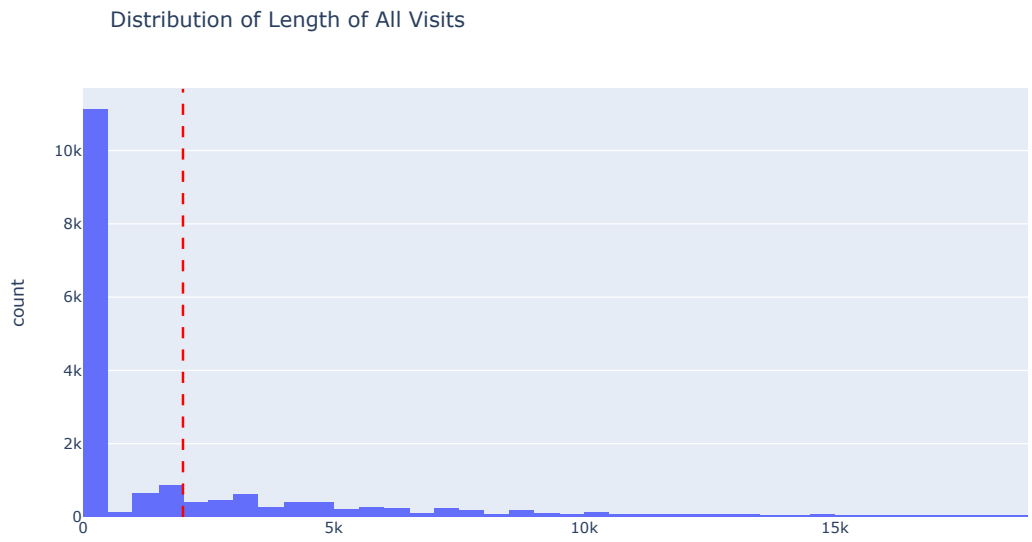
Questions we are curious about:

- What is the average time people tend to stay at the park?
- Are cars speeding?

```
# convert Timestamp to datetime values, in case it is not already
df['Timestamp'] = pd.to_datetime(df['Timestamp'])

# visits holds a dictionary with car IDs for keys and lists for values, where the first value in the inner list is the entrance and the sec
visits = []
visitLengths = []
# looping through each unique car-id
for id in df['car-id'].unique():

  # limiting the dataframe to only be relevant to the car we're looking at
  curr_df = df.loc[(df['car-id'] == id) & (df['gate-type'] == 'entrance')]
  # count to show if on an exit or an entrance, where odd numbers show an entrance
  entrance = True
  # for loop to loop through the df
  newVisit = []
  length = 0
```

```
  for index, row in curr_df.iterrows():
    if (entrance):
      newVisit.append(row['Timestamp'])
      entrance = False
    else:
      newVisit.append(row['Timestamp'])
      # calculate length of the visit we are on
      length = (newVisit[1] - newVisit[0]).total_seconds()/60
      visitLengths.append(round(length))
      visits.append([id, newVisit, length])
      newVisit = []
      entrance = True


visits_df = pd.DataFrame(visitLengths, columns = ['visit_length'])
```

```
visits_df = visits_df.sort_values(by=['visit_length'])
fig = px.histogram(visits_df, x="visit_length", nbins = 150, title = 'Distribution of Length of All Visits')
fig.add_vline(x=visits_df['visit_length'].mean(), line_dash="dash", line_color="red")
fig.show()
```




```
short_df = visits_df[visits_df['visit_length'] <= visits_df['visit_length'].mean()]
fig = px.histogram(short_df, x = 'visit_length', nbins = 200, title = 'Distribution of Length of Visits (0-200 minutes)')
fig.add_vrect(x0 = 0, x1 = 28.8, fillcolor="red", opacity=0.3)
fig.update_xaxes(range=[0,200])
fig.show()
```

Analysis of above Figures

- Figure 1 -

The first figure 'Distribution of Length of All Visits' is a histogram of the visit length data. The red vertical dotted line represents the mean length of all visits.

The interactivity in this graph allows users to hover over each bin and see the exact count of visits and the range of length of visits.

Observations The mode of all visit lengths falls around 8 hours indicating that day-length trips and shorter are most common. There are also outliers that visit the park for a month, which are likely explained by rangers patrolling and sharing cars when swapping out at ranger stations.

- Figure 2 -

The second figure 'Distribution of Length of Visits (0-200 minutes) is a zoom in on the first figure's cluster of visits between 0-200 minutes. This graph has the same interactivity as Fig 1 with the hover tool.

Overall Observations of Patterns & Irregularities

Expectations Cars would be expected to follow the speed limit of 25 mph which means traveling approximately 12 miles from entrance to entrance would take approximately 29 minutes.

Irregularities The red box in the second graph overlays the range of visit times under 29 minutes which indicates the driver was speeding through the preserve to get through in that amount of time.

*****

POTENTIAL IMPACT #4

Disturbances from speeding vehicles: *Thousands of cars pass through the preserve in less than 29 minutes, the estimated amount of time the drive through the preserve should take if a car is following the speed limit. This speeding could result in reckless endangerment of the local wildlife because drivers do not have enough reaction time to stop in case of animals crossing the road. This can also result in excess emissions and noise pollution that is unnatural to the wildlife habitat meant to be protected by the preserve.*

*****

Build hypothesis on what the patterns represent

Overall, the top three potential impacts on the species in the park are:

1. High periods of activity in the park resulting in pollution/disturbances
2. Pollution from large vehicles venturing in restricted areas
3. Disturbances from speeding vehicles

Information That Could Help Future Analysis

- More detailed car-type tracking (are car-types 4 commercial vehicles or normal cars trailing a trailor?)
- Looking specifically into the paths taken by each car and seeing which is most popular and measuring distance of those paths specifically
- Getting information on species life in the park to see if consequential impacts to species line up to certain vehicle activity