UFO Data Science Project

Antonella Basso

0.1 UFO Data

Source: https://www.kaggle.com/NUFORC/ufo-sightings

Columns:

- datetime: date and time of the sighting (most time in military time)
- city: name of the city where the sighting took place
- state: name (abreviated) of the state (in the US) where the sighting took place
- country: name (abreviated) of the country where the sighting took place
- shape: shape of the sighting
- duration (seconds): duration of the sighting in seconds
- duration (hours/min): duration of the sighting in hours and minutes
- comments: breif comment providing more information about the sighting
- date posted: date the sighting was reported/posted
- latitude: latitude of location of sighting
- longitude: longitude of location of sighting

Goals/Exploration:

- Visual representation of the different shapes of the sightings in the US
- Visual representation of the number of sightings that took place in each state in the US
- Linear regression of the number of sightings that took place at each hour of the day
- Creating a choropleth map of the US to show where the sightings take place

[195]:
```
p
```

[195]:
```
['08D',
 '1506-',
 '1507-',
 '1508-',
 '1509-',
 '1510-',
 '2015-',
```

```
'25',
'2R-OW',
'31',
'32',
'34',
'36',
'37',
'7',
'8',
'93100',
'94126',
'94128',
'94139',
'96153',
'96154',
'96156',
'AP-05',
'AP-5',
'APW5',
'APW6',
'AW-08',
'B-123',
'B-126',
'B-26',
'BA05',
'BA06',
'BCC-M',
'BMW-1',
'BMW-2',
'BMW-3',
'BW-1S',
'BW-2S',
'BW-3D',
'Blue',
'C-18',
'C-19',
'C-4',
'CA-06',
'CCR-B',
'EBG',
'F-3A',
'F-4',
'G01D',
'G02D',
'G02S',
'G03D',
'G04S',
```

```
'G101',
'G102',
'G200',
'G201',
'G270',
'G280',
'G281',
'G306',
'G45S',
'G48MG',
'G50S',
'G51S',
'H-2',
'H-3',
'H-4',
'H2OX',
'H3OX',
'H4OX',
'IMW-1',
'IMW-2',
'JTEC-',
'KC-15',
'M-12',
'M-6',
'MGS-5',
'MW 6',
'MW-01',
'MW-03',
'MW-04',
'MW-06',
'MW-08',
'MW-09',
'MW-1',
'MW-10',
'MW-17',
'MW-24',
'MW-27',
'MW-29',
'MW-2R',
'MW-3',
'MW-31',
'MW-3D',
'MW-4',
'MW-42',
'MW-4D',
'MW-4I',
'MW-4N',
```

```
'MW-4R',
'MW-4S',
'MW-5C',
'MW-70',
'MW-72',
'MW-75',
'MW-84',
'MW-8I',
'MW-8S',
'MW-9D',
'MW-9I',
'MW-9R',
'MW-9S',
'MW-B1',
'MW-B2',
'MW-DP',
'MW-FG',
'MW-N0',
'MW201',
'MW203',
'MW24D',
'MW85',
'MW86',
'MW87',
'OAE-2',
'OW-45',
'OW-46',
'P-125',
'P-130',
'P-131',
'P-134',
'P-141',
'P-150',
'P-151',
'P-152',
'P-153',
'P-154',
'P-157',
'P-158',
'P-17',
'P-23',
'P-6',
'PONDN',
'R201',
'S-17',
'S-19A',
'S-2',
```

```
 'S-25',
 'S-4',
 'S-5',
 'S-6',
 'SGS-3',
 'SLAG8',
 'SLAG9',
 'T03S',
 'TP-20',
 'W-101',
 'W-102',
 'W-107',
 'W20D',
 'W46D',
 'W48',
 'W77',
 'WBSP-']
```

[195]:
```
p
```

[195]:
```
['08D',
 '1506-',
 '1507-',
 '1508-',
 '1509-',
 '1510-',
 '2015-',
 '25',
 '2R-OW',
 '31',
 '32',
 '34',
 '36',
 '37',
 '7',
 '8',
 '93100',
 '94126',
 '94128',
 '94139',
 '96153',
 '96154',
 '96156',
 'AP-05',
 'AP-5',
 'APW5',
 'APW6',
```

```
'AW-08',
'B-123',
'B-126',
'B-26',
'BA05',
'BA06',
'BCC-M',
'BMW-1',
'BMW-2',
'BMW-3',
'BW-1S',
'BW-2S',
'BW-3D',
'Blue',
'C-18',
'C-19',
'C-4',
'CA-06',
'CCR-B',
'EBG',
'F-3A',
'F-4',
'G01D',
'G02D',
'G02S',
'G03D',
'G04S',
'G101',
'G102',
'G200',
'G201',
'G270',
'G280',
'G281',
'G306',
'G45S',
'G48MG',
'G50S',
'G51S',
'H-2',
'H-3',
'H-4',
'H2OX',
'H3OX',
'H4OX',
'IMW-1',
'IMW-2',
```

```
'JTEC-',
'KC-15',
'M-12',
'M-6',
'MGS-5',
'MW 6',
'MW-01',
'MW-03',
'MW-04',
'MW-06',
'MW-08',
'MW-09',
'MW-1',
'MW-10',
'MW-17',
'MW-24',
'MW-27',
'MW-29',
'MW-2R',
'MW-3',
'MW-31',
'MW-3D',
'MW-4',
'MW-42',
'MW-4D',
'MW-4I',
'MW-4N',
'MW-4R',
'MW-4S',
'MW-5C',
'MW-70',
'MW-72',
'MW-75',
'MW-84',
'MW-8I',
'MW-8S',
'MW-9D',
'MW-9I',
'MW-9R',
'MW-9S',
'MW-B1',
'MW-B2',
'MW-DP',
'MW-FG',
'MW-N0',
'MW201',
'MW203',
```

```
 'MW24D',
 'MW85',
 'MW86',
 'MW87',
 'OAE-2',
 'OW-45',
 'OW-46',
 'P-125',
 'P-130',
 'P-131',
 'P-134',
 'P-141',
 'P-150',
 'P-151',
 'P-152',
 'P-153',
 'P-154',
 'P-157',
 'P-158',
 'P-17',
 'P-23',
 'P-6',
 'PONDN',
 'R201',
 'S-17',
 'S-19A',
 'S-2',
 'S-25',
 'S-4',
 'S-5',
 'S-6',
 'SGS-3',
 'SLAG8',
 'SLAG9',
 'T03S',
 'TP-20',
 'W-101',
 'W-102',
 'W-107',
 'W20D',
 'W46D',
 'W48',
 'W77',
 'WBSP-']
```

[1]:
```
import ast
import pandas as pd
```

```
import numpy as np
import altair as alt
from altair import Chart, X, Y, Color, Scale
```

[2]:
```
df = pd.read_csv('ufosightings.csv').dropna(axis=1, how='all')
df.head()
```

[2]:
```
            datetime                  city state country     shape  \
0  10/10/1949 20:30            san marcos    tx      us  cylinder
1  10/10/1949 21:00          lackland afb    tx     NaN     light
2  10/10/1955 17:00  chester (uk/england)   NaN      gb    circle
3  10/10/1956 21:00                  edna    tx      us    circle
4  10/10/1960 20:00               kaneohe    hi      us     light

  duration (seconds) duration (hours/min)  \
0               2700           45 minutes
1               7200              1-2 hrs
2                 20           20 seconds
3                 20             1/2 hour
4                900           15 minutes

                                          comments date posted    latitude  \
0  This event took place in early fall around 194...   4/27/2004  29.8830556
1  1949 Lackland AFB&#44 TX.  Lights racing acros...  12/16/2005    29.38421
2  Green/Orange circular disc over Chester&#44 En...   1/21/2008        53.2
3  My older brother and twin sister were leaving ...   1/17/2004  28.9783333
4  AS a Marine 1st Lt. flying an FJ4B fighter/att...   1/22/2004  21.4180556

    longitude  Unnamed: 11
0  -97.941111          NaN
1  -98.581082          NaN
2   -2.916667          NaN
3  -96.645833          NaN
4 -157.803611          NaN
```

0.1.1 Data Cleanup:

- Elimination of other coutries besides the US for the purpose of condensing the data
- Elimination of the 11th column containing NaN and any rows containing NaN
- Combining data in the "shape" column by replacing outliers such as "delta" with more standard shape names like "triangle" (this is something I really wanted to do with my data before and was not able to)

[3]:
```
us_df = df[(df.country=='us')].reset_index()
us_df.head()
```

[3]:
```
   index          datetime        city state country     shape  \
0      0  10/10/1949 20:30  san marcos    tx      us  cylinder
1      3  10/10/1956 21:00        edna    tx      us    circle
2      4  10/10/1960 20:00     kaneohe    hi      us     light
3      5  10/10/1961 19:00     bristol    tn      us    sphere
4      7  10/10/1965 23:45     norwalk    ct      us      disk

  duration (seconds) duration (hours/min)  \
0               2700           45 minutes
1                 20             1/2 hour
2                900           15 minutes
3                300            5 minutes
4               1200           20 minutes

                                            comments date posted    latitude  \
0  This event took place in early fall around 194...   4/27/2004  29.8830556
1  My older brother and twin sister were leaving ...   1/17/2004  28.9783333
2  AS a Marine 1st Lt. flying an FJ4B fighter/att...   1/22/2004  21.4180556
3  My father is now 89 my brother 52 the girl wit...   4/27/2007      36.595
4  A bright orange color changing to reddish colo...   10/2/1999     41.1175

    longitude  Unnamed: 11
0  -97.941111          NaN
1  -96.645833          NaN
2 -157.803611          NaN
3  -82.188889          NaN
4  -73.408333          NaN
```

[4]:
```
len(us_df)
```

[4]:
```
51880
```

[5]:
```
us_df.shape
```

[5]:
```
(51880, 13)
```

[6]:
```
us_df = us_df.dropna(axis=1, how='all')
us_df.head()
```

[6]:
```
   index          datetime        city state country     shape  \
0      0  10/10/1949 20:30  san marcos    tx      us  cylinder
1      3  10/10/1956 21:00        edna    tx      us    circle
2      4  10/10/1960 20:00     kaneohe    hi      us     light
3      5  10/10/1961 19:00     bristol    tn      us    sphere
4      7  10/10/1965 23:45     norwalk    ct      us      disk

  duration (seconds) duration (hours/min)  \
```

```
0               2700           45 minutes
1                 20             1/2 hour
2                900           15 minutes
3                300            5 minutes
4               1200           20 minutes

                                          comments date posted    latitude  \
0  This event took place in early fall around 194...   4/27/2004  29.8830556
1  My older brother and twin sister were leaving ...   1/17/2004  28.9783333
2  AS a Marine 1st Lt. flying an FJ4B fighter/att...   1/22/2004  21.4180556
3  My father is now 89 my brother 52 the girl wit...   4/27/2007      36.595
4  A bright orange color changing to reddish colo...   10/2/1999     41.1175

    longitude
0  -97.941111
1  -96.645833
2 -157.803611
3  -82.188889
4  -73.408333
```

[7]:
```
us_df.shape
```

[7]:
```
(51880, 12)
```

[8]:
```
us_df = us_df.dropna()
us_df.head()
```

[8]:
```
   index          datetime        city state country     shape  \
0      0  10/10/1949 20:30  san marcos    tx      us  cylinder
1      3  10/10/1956 21:00        edna    tx      us    circle
2      4  10/10/1960 20:00     kaneohe    hi      us     light
3      5  10/10/1961 19:00     bristol    tn      us    sphere
4      7  10/10/1965 23:45     norwalk    ct      us      disk

  duration (seconds) duration (hours/min)  \
0               2700           45 minutes
1                 20             1/2 hour
2                900           15 minutes
3                300            5 minutes
4               1200           20 minutes

                                          comments date posted    latitude  \
0  This event took place in early fall around 194...   4/27/2004  29.8830556
1  My older brother and twin sister were leaving ...   1/17/2004  28.9783333
2  AS a Marine 1st Lt. flying an FJ4B fighter/att...   1/22/2004  21.4180556
3  My father is now 89 my brother 52 the girl wit...   4/27/2007      36.595
4  A bright orange color changing to reddish colo...   10/2/1999     41.1175
```

```
    longitude
0  -97.941111
1  -96.645833
2 -157.803611
3  -82.188889
4  -73.408333
```

[9]:
```
us_df.shape
```

[9]:
```
(48893, 12)
```

[10]:
```
shapes = pd.Series(us_df['shape'])
shapes
```

[10]:
```
0         cylinder
1           circle
2            light
3           sphere
4             disk
5             disk
6             disk
7           circle
8         fireball
9             disk
10         unknown
11            oval
12          circle
13            disk
14            disk
15           other
16           light
17           light
18            oval
19           other
20           light
21       rectangle
22         chevron
23            oval
24         unknown
25          sphere
26         unknown
27           light
28          circle
29           other
           ...
51848     triangle
```

```
51849     teardrop
51850        light
51851        light
51852     teardrop
51853     fireball
51855        light
51856       circle
51857      unknown
51858       sphere
51859        flash
51860        other
51862     fireball
51863     triangle
51864     triangle
51865         disk
51866         disk
51867     fireball
51868     triangle
51869        light
51870        light
51871       circle
51872       sphere
51873     fireball
51874        light
51875     triangle
51876     fireball
51877         oval
51878        light
51879       circle
Name: shape, Length: 48893, dtype: object
```

[11]:
```
shapes = shapes.replace(to_replace=r'^delta', value='triangle', regex=True)
```

[12]:
```
shapes = shapes.replace({'changed': 'changing'})
```

[13]:
```
shapes = shapes.replace({'round': 'circle'})
```

[14]:
```
shapes.head()
```

[14]:
```
0    cylinder
1      circle
2       light
3      sphere
4        disk
Name: shape, dtype: object
```

[15]:
```
from collections import Counter
Counter(shapes)
```

[15]:
```
Counter({'cylinder': 782,
         'circle': 4661,
         'light': 10175,
         'sphere': 3285,
         'disk': 3264,
         'fireball': 3706,
         'unknown': 3653,
         'oval': 2377,
         'other': 3545,
         'rectangle': 790,
         'chevron': 660,
         'formation': 1593,
         'triangle': 5114,
         'cigar': 1227,
         'changing': 1267,
         'diamond': 743,
         'flash': 787,
         'egg': 467,
         'teardrop': 445,
         'cone': 192,
         'cross': 156,
         'pyramid': 1,
         'flare': 1,
         'hexagon': 1,
         'crescent': 1})
```

0.2 Charts:

1. Looking at the different shapes of the sightings recorded in the US and comparing the count of each shape in a bar graph

[16]:
```
shapes_dict = dict(Counter(shapes))
shapes_dict
```

[16]:
```
{'cylinder': 782,
 'circle': 4661,
 'light': 10175,
 'sphere': 3285,
 'disk': 3264,
 'fireball': 3706,
 'unknown': 3653,
 'oval': 2377,
 'other': 3545,
 'rectangle': 790,
```

```
 'chevron': 660,
 'formation': 1593,
 'triangle': 5114,
 'cigar': 1227,
 'changing': 1267,
 'diamond': 743,
 'flash': 787,
 'egg': 467,
 'teardrop': 445,
 'cone': 192,
 'cross': 156,
 'pyramid': 1,
 'flare': 1,
 'hexagon': 1,
 'crescent': 1}
```

[17]:
```
us_shapes = pd.Series(shapes_dict)
us_shapes.head()
```

[17]:
```
cylinder      782
circle       4661
light       10175
sphere       3285
disk         3264
dtype: int64
```

[18]:
```
us_shapes1 = pd.DataFrame({'count': us_shapes})
us_shapes1.head()
```

[18]:

	count
cylinder	782
circle	4661
light	10175
sphere	3285
disk	3264

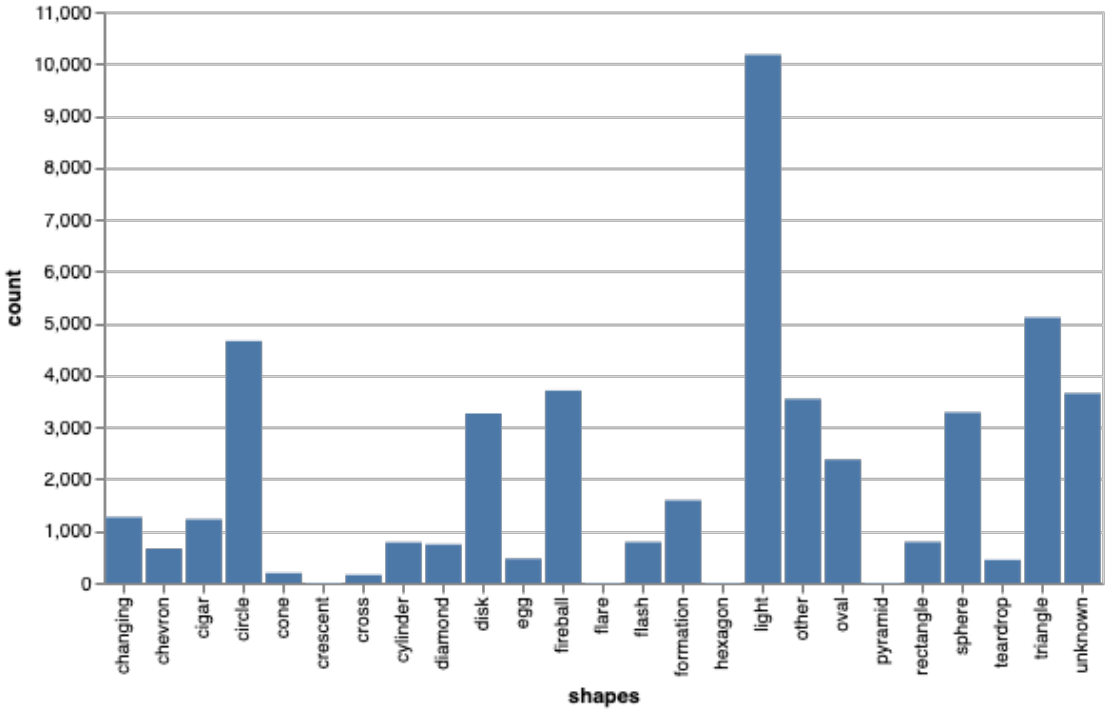
[19]:
```
us_shapes1['shapes'] = us_shapes1.index
us_shapes1.head()
```

[19]:

	count	shapes
cylinder	782	cylinder
circle	4661	circle
light	10175	light
sphere	3285	sphere
disk	3264	disk

[20]:
```
alt.Chart(us_shapes1).mark_bar().encode(x='shapes', y='count')
```

[20]:



The goal of this graph is to show the frequency of the shapes of UFO sightings in the US. As can be noticed, common, more general shapes such as ‘‘circle’’ and ‘‘triangle’’ had more occurrences, whereas rare and more descriptive shapes like ‘‘crescent’’ and ‘‘flare’’ had much fewer occurrences. This is likely due to the fact that it is more probable for a group of people to have a generic description in common, than a more specific one. Further, it can be noticed that ‘‘light’’ is by far the most common shape of sighting which is not very surprising because all kinds of light can be frequently seen in the sky, much of which might not necessarily be a UFO. For instance, it is more likely to spot a light source such as a star, planet or satellite in the sky and mistake it with a UFO, than to spot a ‘‘hexagon’’ or ‘‘teardrop’’, which may more likely be so. Another observation that can be made is that ‘‘other’’ and ‘‘unknown’’ also have a relatively large number of occurrences, which too makes sense because as sightings tend to occur so quickly, it may be difficult to classify their shape specifically. Additionally, it is easy to classify very specific shapes as ‘‘other’’ resulting in a high number of occurrences for that category.

2. Making a bar graph to show and compare the number of sightings in each state

[21]:
```
states_count = pd.DataFrame({'count':us_df['state'].count(), 'states':
 ↪us_df['state']})
states_count.head()
```

[21]:

	count	states
0	48893	tx
1	48893	tx
2	48893	hi
3	48893	tn
4	48893	ct

[22]:
```
st = [state for state in us_df['state']]
len(st)
```

[22]: 48893

[23]:
```
states_dict = dict(Counter(st))
states_dict
```

[23]:
```
{'tx': 2730,
 'hi': 207,
 'tn': 897,
 'ct': 634,
 'al': 540,
 'fl': 3199,
 'ca': 6890,
 'nc': 1378,
 'ny': 2156,
 'ky': 635,
 'mi': 1259,
 'ma': 922,
 'ks': 451,
 'sc': 778,
 'wa': 2548,
 'co': 1068,
 'nh': 347,
 'wi': 853,
 'me': 349,
 'ga': 1034,
 'pa': 1772,
 'il': 1743,
 'ar': 464,
 'mo': 1103,
 'oh': 1639,
 'in': 956,
 'az': 1935,
 'mn': 705,
 'nv': 612,
 'ne': 258,
 'or': 1169,
 'ia': 516,
```

```
 'va': 1031,
 'id': 378,
 'nm': 554,
 'nj': 921,
 'wv': 342,
 'ok': 539,
 'ak': 266,
 'ri': 170,
 'vt': 198,
 'la': 460,
 'nd': 79,
 'pr': 22,
 'ms': 313,
 'ut': 472,
 'md': 644,
 'mt': 366,
 'wy': 123,
 'sd': 127,
 'de': 138,
 'dc': 3}
```

[24]:
```
us_states = pd.Series(states_dict)
us_states.head()
```

[24]:
```
tx    2730
hi     207
tn     897
ct     634
al     540
dtype: int64
```

[25]:
```
us_states1 = pd.DataFrame({'count': us_states})
us_states1.head()
```

[25]:
```
    count
tx   2730
hi    207
tn    897
ct    634
al    540
```

[26]:
```
us_states1['states'] = us_states1.index
us_states1.head()
```

[26]:
```
    count states
tx   2730     tx
hi    207     hi
```

```
tn    897    tn
ct    634    ct
al    540    al
```

[27]:
```
len(us_states1)
```

[27]: 52

[28]:
```
us_states1 = us_states1.drop('pr', axis=0)
```

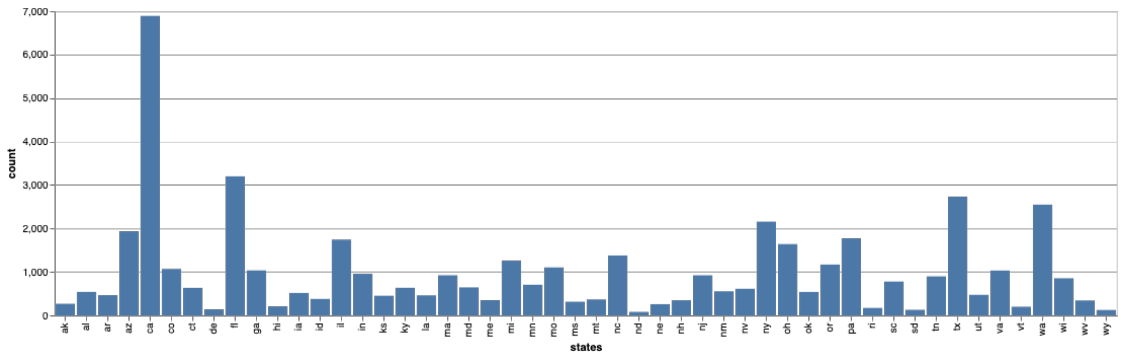
[29]:
```
us_states1 = us_states1.drop('dc', axis=0)
```

[30]:
```
len(us_states1)
```

[30]: 50

[31]:
```
alt.Chart(us_states1).mark_bar().encode(x='states', y='count')
```

[31]:



The goal of this graph is to show the number of sightings per state in the US and compare them visually. As can be seen, California has by far the most number of sightings with more that double the amount of any other state (almost 7,000). This may be due to the fact that California is a large state and is highly populated. Further, it could have something to do with the fact that it borders the Pacific Ocean, but that claim can't be made with certainty. Other large states such as Florida and Texas had a noticeably large number of sightings as well, which may also be a result of their size and population.

3. Using Linear Regression to observe the correlation between the number of sightings and the time of day they occured and predict how many sightings may occur at a specific hour during the day (x = time of day, y = number of sightings)

[32]:
```
datetime = pd.Series(us_df[str('datetime')])
datetime.head()
```

[32]:
```
0    10/10/1949 20:30
1    10/10/1956 21:00
2    10/10/1960 20:00
3    10/10/1961 19:00
4    10/10/1965 23:45
Name: datetime, dtype: object
```

[33]:
```
datetime = pd.DataFrame({'time': us_df[str('datetime')]})
datetime.head()
```

[33]:
```
              time
0  10/10/1949 20:30
1  10/10/1956 21:00
2  10/10/1960 20:00
3  10/10/1961 19:00
4  10/10/1965 23:45
```

[34]:
```
time = datetime.astype(str).apply(lambda x: x.str[11:13])
time.head()
```

[34]:
```
  time
0   20
1   21
2   20
3   19
4   23
```

[35]:
```
time = time.replace({'1:': '01'})
```

[36]:
```
time = time.replace({'2:': '02'})
```

[37]:
```
time = time.replace({'3:': '03'})
```

[38]:
```
time = time.replace({'4:': '04'})
```

[39]:
```
time = time.replace({'5:': '05'})
```

[40]:
```
time = time.replace({'6:': '06'})
```

[41]:
```
time = time.replace({'7:': '07'})
```

[42]:
```
time = time.replace({'8:': '08'})
```

[43]:
```
time = time.replace({'9:': '09'})
```

[44]:
```
time = time.replace({'0:': '10'})
```

[45]:
```
time = time.replace({'00': '24'})
```

[46]:
```
Counter(time['time'])
```

[46]:
```
Counter({'20': 1494,
         '21': 1308,
         '19': 1427,
         '23': 910,
         '13': 154,
         '16': 330,
         '22': 1111,
         '17': 617,
         '12': 149,
         '02': 5242,
         '24': 506,
         '05': 1344,
         '18': 1138,
         '03': 4050,
         '04': 1450,
         '06': 1210,
         '11': 121,
         '08': 1736,
         '14': 151,
         '15': 210,
         '07': 1365,
         '09': 2590,
         '01': 5898,
         '10': 4906,
         ':0': 4633,
         ':3': 2001,
         ':5': 486,
         ':4': 915,
         ':1': 924,
         ':2': 517})
```

[47]:
```
time_dict = dict(Counter(time['time']))
time_dict
```

[47]:
```
{'20': 1494,
 '21': 1308,
 '19': 1427,
 '23': 910,
 '13': 154,
 '16': 330,
 '22': 1111,
 '17': 617,
 '12': 149,
```

```
 '02': 5242,
 '24': 506,
 '05': 1344,
 '18': 1138,
 '03': 4050,
 '04': 1450,
 '06': 1210,
 '11': 121,
 '08': 1736,
 '14': 151,
 '15': 210,
 '07': 1365,
 '09': 2590,
 '01': 5898,
 '10': 4906,
 ':0': 4633,
 ':3': 2001,
 ':5': 486,
 ':4': 915,
 ':1': 924,
 ':2': 517}
```

[48]:
```
us_time = pd.Series(time_dict)
us_time
```

[48]:
```
20    1494
21    1308
19    1427
23     910
13     154
16     330
22    1111
17     617
12     149
02    5242
24     506
05    1344
18    1138
03    4050
04    1450
06    1210
11     121
08    1736
14     151
15     210
07    1365
09    2590
```

```
01    5898
10    4906
:0    4633
:3    2001
:5     486
:4     915
:1     924
:2     517
dtype: int64
```

[49]:
```
us_time1 = pd.DataFrame({'count': us_time})
us_time1.head()
```

[49]:
```
    count
20   1494
21   1308
19   1427
23    910
13    154
```

[50]:
```
us_time1['time'] = us_time1.index
us_time1.head()
```

[50]:
```
    count time
20   1494   20
21   1308   21
19   1427   19
23    910   23
13    154   13
```

[51]:
```
us_time1 = us_time1.drop(':0', axis=0)
```

[52]:
```
us_time1 = us_time1.drop(':1', axis=0)
```

[53]:
```
us_time1 = us_time1.drop(':2', axis=0)
```

[54]:
```
us_time1 = us_time1.drop(':3', axis=0)
```

[55]:
```
us_time1 = us_time1.drop(':4', axis=0)
```

[56]:
```
us_time1 = us_time1.drop(':5', axis=0)
```

[57]:
```
len(us_time1)
```

[57]:
```
24
```

[58]:
```
us_time1.head()
```

[58]:

	count	time
20	1494	20
21	1308	21
19	1427	19
23	910	23
13	154	13

[59]:
```
us_time1 = us_time1.reset_index()
us_time1.head()
```

[59]:

	index	count	time
0	20	1494	20
1	21	1308	21
2	19	1427	19
3	23	910	23
4	13	154	13

[60]:
```
us_time1 = us_time1.drop("index", axis=1)
us_time1.head()
```

[60]:

	count	time
0	1494	20
1	1308	21
2	1427	19
3	910	23
4	154	13

[61]:
```
us_time1_int = us_time1.astype(int)
us_time1_int.head()
```

[61]:

	count	time
0	1494	20
1	1308	21
2	1427	19
3	910	23
4	154	13

[62]:
```
us_time1_int = us_time1_int.sort_values(by='time', ascending=True)
us_time1_int.head()
```

[62]:

	count	time
22	5898	1
9	5242	2
13	4050	3
14	1450	4
11	1344	5

[63]:
```
us_time1_int = us_time1_int.reset_index()
us_time1_int.head()
```

[63]:

	index	count	time
0	22	5898	1
1	9	5242	2
2	13	4050	3
3	14	1450	4
4	11	1344	5

[64]:
```
us_time1_int = us_time1_int.drop("index", axis=1)
us_time1_int.head()
```

[64]:

	count	time
0	5898	1
1	5242	2
2	4050	3
3	1450	4
4	1344	5

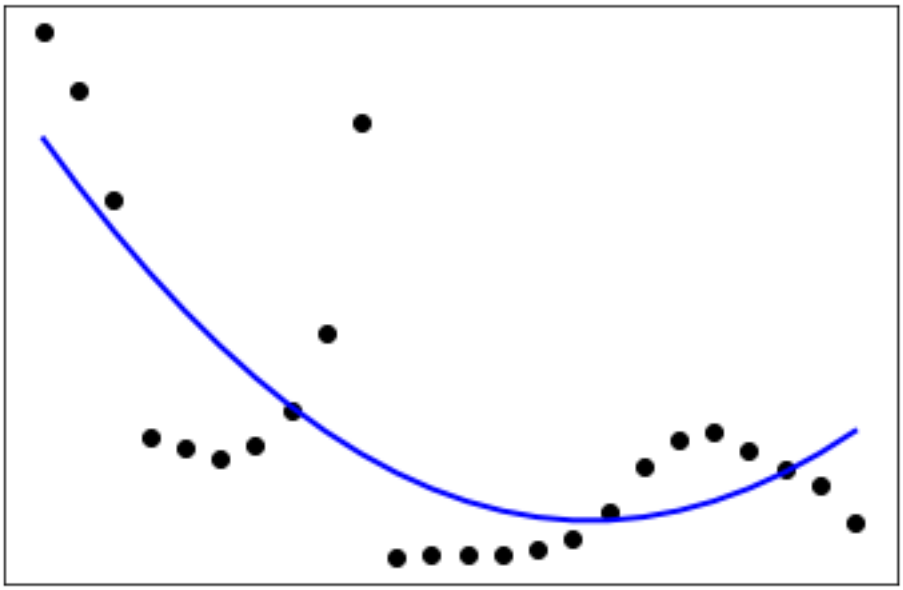
[65]:
```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures

x = us_time1_int['time']
y = us_time1_int['count']
x, y = np.array(x).reshape(-1, 1), np.array(y)

x_ = PolynomialFeatures(degree=2, include_bias=False).fit_transform(x)

model = LinearRegression().fit(x_, y)

r_sq = model.score(x_, y)
intercept, coefficients = model.intercept_, model.coef_

y_pred = model.predict(x_)
```

[66]:
```
print('coefficient of determination:', r_sq)
print('intercept:', intercept)
print('coefficients:', coefficients, sep='\n')
print('predicted response:', y_pred, sep='\n')
```

```
coefficient of determination: 0.5459176544714863
intercept: 5269.420454545456
coefficients:
[-574.84819854   17.42966707]
predicted response:
[4712.00192308 4189.44272575 3701.74286257 3248.90233354 2830.92113864
```

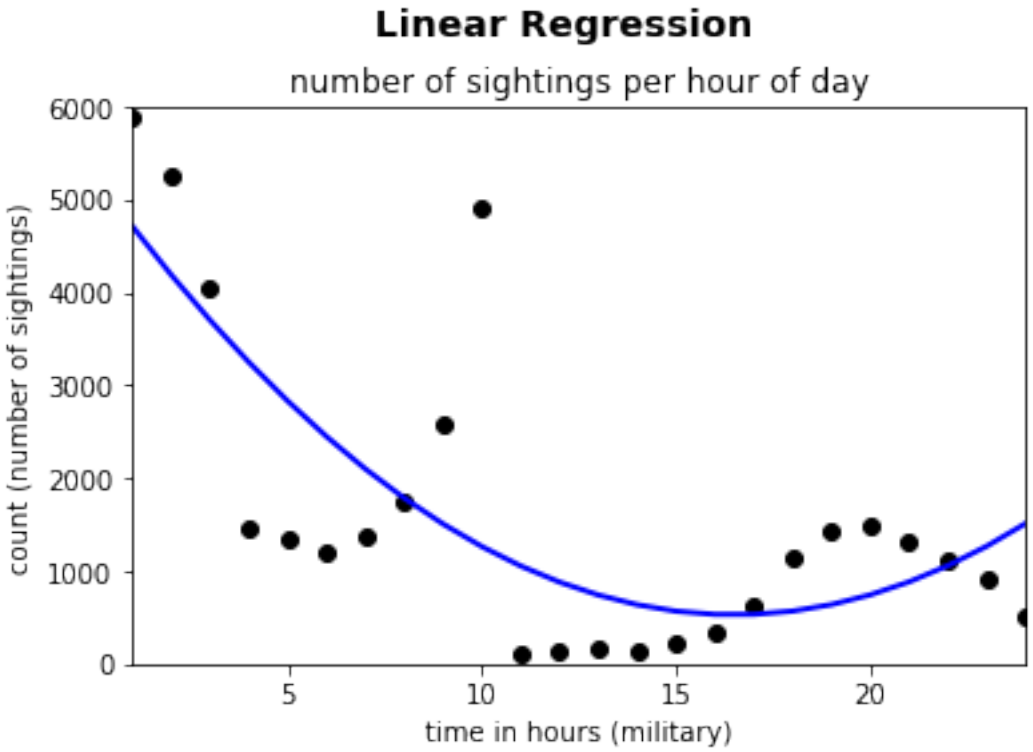
```
 2447.7992779  2099.53675129 1786.13355883 1507.58970052 1263.90517635
 1055.07998632  881.11413043  742.0076087   637.7604211   568.37256765
  533.84404834  534.17486318  569.36501216  639.41449529  744.32331256
  884.09146397 1058.71894953 1268.20576923 1512.55192308]
```

[81]:
```
import matplotlib.pyplot as plt

splot = plt.scatter(x, y, color='black')
pplot = plt.plot(x, y_pred, color='blue', linewidth=2)

plt.xticks(())
plt.yticks(())

plt.show()
```



[68]:
```
fig = plt.figure()
fig.suptitle('Linear Regression', fontsize=14, fontweight='bold')

ax = fig.add_subplot(111)
fig.subplots_adjust(top=0.85)
ax.set_title('number of sightings per hour of day')

ax.set_xlabel('time in hours (military)')
ax.set_ylabel('count (number of sightings)')

plt.scatter(x, y, color='black')
plt.plot(x, y_pred, color='blue', linewidth=2)
```

```
plt.xticks()
plt.yticks()

ax.axis([1, 24, 0, 6000])
plt.show()
```



The goal of this linear regression is to show a correlation between the frequency of sightings and the time of occurence (time of day in hours), and graph a prediction of how many sightings may occur given a specific hour (time) in the day. Due to the fact that this relationship is not at all linear, a polynomial with degree 2 (quadratic) was used to make a more accurate approximation. The graph above shows the number of sightings (y-axis) that occured at every hour of the day (military time), starting with 1am and ending at the 24th hour (12am) along the x-axis. As can be noticed, most sightings occurred in the middle of the night, between 1am and 3am, which is when the sky is darkest. This makes sense because light is easier to see when the sky is as dark as possible. Moreover, there are two more peaks, specifically at around sunrise (10am) and sunset (10pm). One interesting point to make regarding this observation is that the planet Venus can be seen at sunrise and it so happens to be the case that this planet is easily mistaken for a UFO.

Choropleth:

[69]:
```
us_states1 = us_states1.reset_index()
us_states1.head()
```

[69]:

	index	count	states
0	tx	2730	tx
1	hi	207	hi
2	tn	897	tn
3	ct	634	ct
4	al	540	al

[70]:
```
us_states1 = us_states1.drop("index", axis=1)
us_states1.head()
```

[70]:

	count	states
0	2730	tx
1	207	hi
2	897	tn
3	634	ct
4	540	al

[71]:
```
pd.to_numeric(us_states1['count'])
```

[71]:
```
0     2730
1      207
2      897
3      634
4      540
5     3199
6     6890
7     1378
8     2156
9      635
10    1259
11     922
12     451
13     778
14    2548
15    1068
16     347
17     853
18     349
19    1034
20    1772
21    1743
22     464
23    1103
24    1639
```

```
25     956
26    1935
27     705
28     612
29     258
30    1169
31     516
32    1031
33     378
34     554
35     921
36     342
37     539
38     266
39     170
40     198
41     460
42      79
43     313
44     472
45     644
46     366
47     123
48     127
49     138
Name: count, dtype: int64
```

[72]:
```
us_states1.head()
```

[72]:
```
   count states
0   2730     tx
1    207     hi
2    897     tn
3    634     ct
4    540     al
```

[73]:
```
us_states1 = us_states1.sort_values(by=['states'])
us_states1.head()
```

[73]:
```
    count states
38    266     ak
4     540     al
22    464     ar
26   1935     az
6    6890     ca
```

[75]:
```
us_states1 = us_states1.reset_index()
us_states1.head()
```

[75]:

	level_0	index	count	states
0	0	38	266	ak
1	1	4	540	al
2	2	22	464	ar
3	3	26	1935	az
4	4	6	6890	ca

[76]:
```
us_states1 = us_states1.drop("index", axis=1)
us_states1.head()
```

[76]:

	level_0	count	states
0	0	266	ak
1	1	540	al
2	2	464	ar
3	3	1935	az
4	4	6890	ca

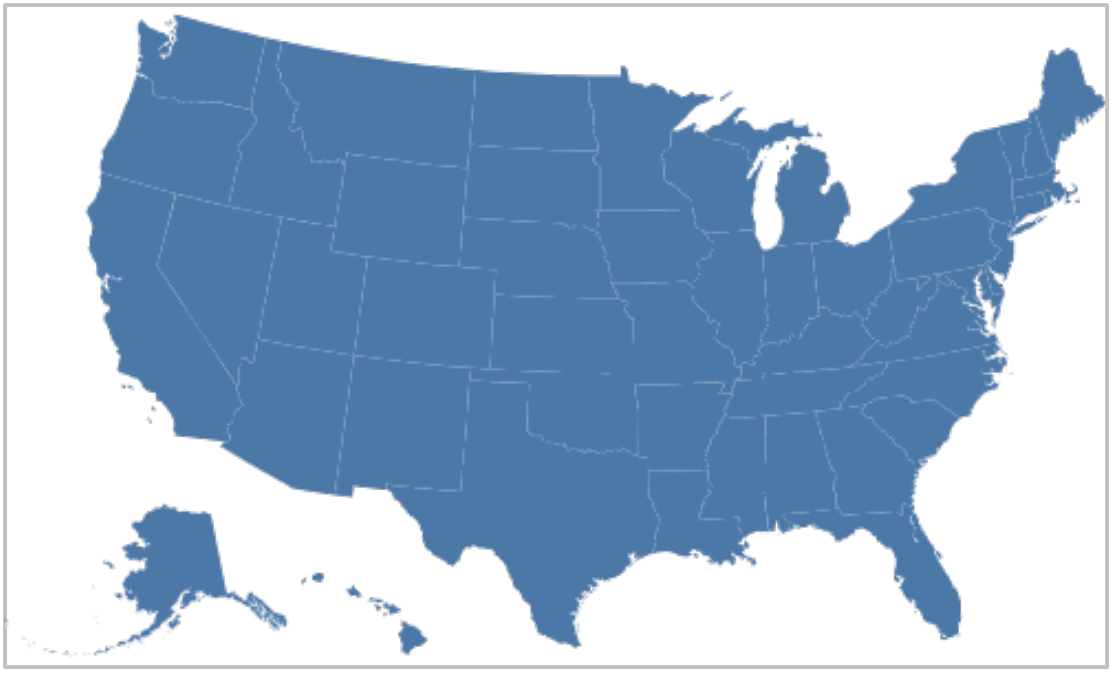
[77]:
```
us_states1 = us_states1.drop("level_0", axis=1)
us_states1.head()
```

[77]:

	count	states
0	266	ak
1	540	al
2	464	ar
3	1935	az
4	6890	ca

[78]:
```
from vega_datasets import data
states = alt.topo_feature(data.us_10m.url, 'states')
data = data.obesity()


alt.Chart(states).mark_geoshape().project(
    type='albersUsa').properties(width=500,height=300)
```

[78]:



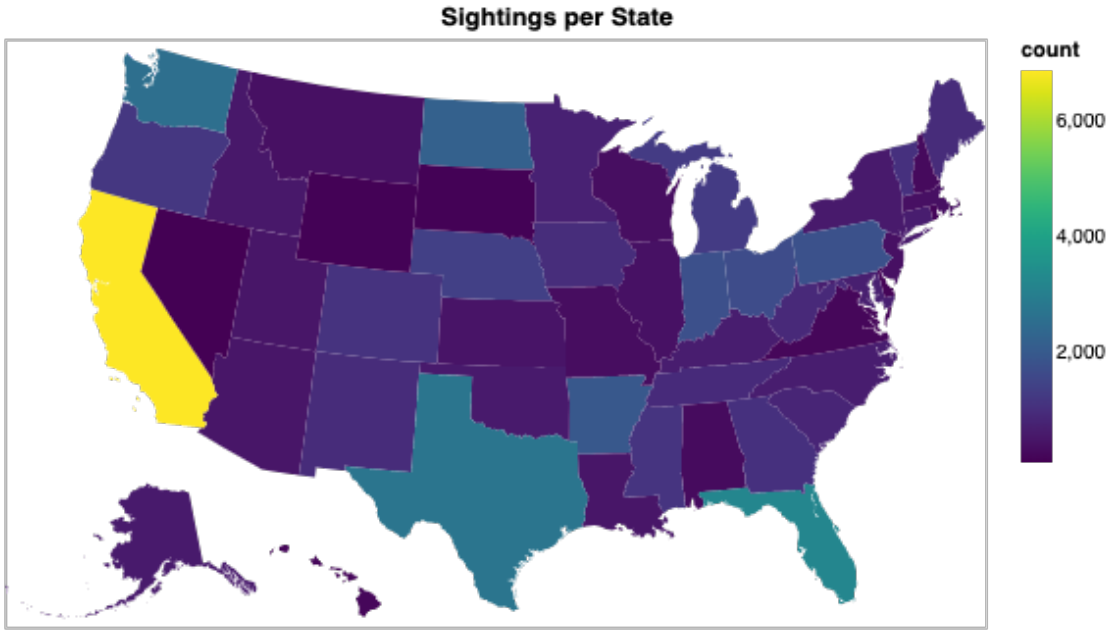
[79]:
```
us_states1['id'] = data['id']
us_states1.head()
```

[79]:

	count	states	id
0	266	ak	1
1	540	al	2
2	464	ar	4
3	1935	az	5
4	6890	ca	6

[80]:
```
alt.Chart(states).mark_geoshape().encode(color='count:Q', tooltip=['states:N']).
↪transform_lookup(lookup='id', from_=alt.LookupData(us_states1, 'id',␣
↪['count'])).project(type='albersUsa').properties(width=500, height=300,␣
↪title='Sightings per State')
```

[80]:



This Choropleth does a better job of visually representing the information that can be obtained from the second bar graph above. While the bar graph is successful in showing the drastic difference between the number of sightings in California in comparison to other states, this graph does a much better job of comparing the number of sightings reported in each state in the country as a whole. From this visual, it is very easy to see that California takes the lead in the number of reported UFO sightings in the last century, as well as the fact that Texas, Florida, Washington and North Dakota seem to be at the same level in this respect and come in second place. From a data science perspective, this graph is far more engaging and far more representative of the kind of data visualization that is wanted today.

[ ]: