# ufo-cleaning

October 28, 2018

Ideas from Kaggle site: - What areas of the country are most likely to have UFO sightings? - Are there any trends in UFO sightings over time? Do they tend to be clustered or seasonal? - Do clusters of UFO sightings correlate with landmarks, such as airports or government research centers? - What are the most common UFO descriptions?

Potential ideas: - Add weather, population... about the sight? - Military base, airport near the sight?

```
In [28]: %matplotlib inline

         import warnings
         import pandas as pd
         import numpy as np
         import seaborn as sns
```

### 0.0.1 Reading data

```
In [127]: # There are some rows with an extra comma that gave reading error
          # Skipped them ~ 300 rows (almost all are nulls)
          # somehow duration seconds has mixed type float and string
          # -> fixed by using low_memory=False
          df = pd.read_csv("../data/complete.csv", error_bad_lines=False, warn_bad_lines=False
          df.head()
```

```
Out[127]:            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
0  -97.941111
1  -98.581082
2   -2.916667
3  -96.645833
4 -157.803611
```

### Not useful columns

```
In [128]: # date posted seem not useful
          df.drop(columns=["date posted"], inplace=True)

          # Save comment to seperate variable for tf-idf
          comments = df.loc[df["comments"].notna(), "comments"]
          shapes = df.loc[df["shape"].notna(), "shape"]

          df.drop(columns=["comments", "shape"], inplace=True)
```

### Casting column types

```
In [129]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88679 entries, 0 to 88678
Data columns (total 8 columns):
datetime                88679 non-null object
city                    88679 non-null object
state                   81270 non-null object
country                 76314 non-null object
duration (seconds)      88677 non-null object
duration (hours/min)    85660 non-null object
latitude                88679 non-null object
longitude               88679 non-null float64
dtypes: float64(1), object(7)
memory usage: 5.4+ MB
```

```
In [130]: # 1 column has wrong value in latitude
          df = df[df.latitude != '33q.200088']
          df["latitude"] = df.latitude.astype(float, errors="ignore")

In [131]: # this column is the same as duration (seconds)
          df.drop(columns=["duration (hours/min)"], inplace=True)
```

```
         # some field have this weird charactor "`"
         df["duration"] = df["duration (seconds)"].str.replace("`", "").astype(np.float32)
         df.loc[df["duration"].isna(), "duration"] = df["duration"].mean()
         df.drop(columns=["duration (seconds)"], inplace=True)
```

```
In [132]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 88678 entries, 0 to 88678
Data columns (total 7 columns):
datetime     88678 non-null object
city         88678 non-null object
state        81269 non-null object
country      76314 non-null object
latitude     88678 non-null float64
longitude    88678 non-null float64
duration     88678 non-null float32
dtypes: float32(1), float64(2), object(4)
memory usage: 5.1+ MB
```

**Fill NAs**

```
In [133]: df[df.country.isna()].shape
```

```
Out[133]: (12364, 7)
```

```
In [134]: # Infer country from state
          df_filled = df.copy()

          # state to country dictionary
          temp = df[["state", "country"]].dropna().drop_duplicates()
          state_to_country = dict(zip(temp["state"], temp["country"]))

          # fill
          df_filled['country'] = df_filled['country'].fillna(df["state"].map(state_to_country)

          # missing country from ~12k to ~4k
          print(df_filled.country.isna().sum())
          df_filled[df_filled.country.isna()].head()
```

```
4662
```

```
Out[134]:             datetime                                  city state country  \
          18   10/10/1973 23:00                        bermuda nas   NaN     NaN
          36   10/10/1982 07:00              gisborne (new zealand)   NaN     NaN
          58   10/10/1993 03:00                   zlatoust (russia)   NaN     NaN
```

```
69  10/10/1996 20:00  lake macquarie (nsw&#44 australia)   NaN     NaN
76  10/10/1998 02:00                       turin (italy)   NaN     NaN

     latitude   longitude  duration
18  32.364167  -64.678611      20.0
36 -38.662334  178.017649     120.0
58  55.183333   59.650000    1200.0
69 -33.093373  151.588982     300.0
76   0.000000    0.000000      15.0
```

In [135]:
```
# Convert datetime column into datetime objects. Time to separate column.
# Some missing or erroneus dates, so using errors='coerce'
# infer_datetime_format=True makes this much quicker
df_filled['datetime'] = pd.to_datetime(df['datetime'], errors='coerce', infer_datetin

# couldn't convert some datetime
# they're a few so we drop
print(df_filled.datetime.isna().sum())
df.loc[df_filled.datetime.isna()].head()
```

```
1220
```

Out[135]:
```
             datetime                     city state country   latitude  \
166  10/10/2005 24:00                 franklin    in      us  39.480556
316  10/11/1994 24:00  hot springs and custer    sd     NaN  43.431646
417  10/11/2006 24:00                     rome    ny      us  43.212778
487  10/11/2012 24:00   truth or consequences    nm      us  33.128333
567   10/1/1972 24:00               sweet home    or      us  44.397778

      longitude  duration
166  -86.055000       0.0
316 -103.474362       0.0
417  -75.456111     120.0
487 -107.252222       0.0
567 -122.735000       0.0
```

In [136]:
```
df_filled = df_filled.dropna(subset=["datetime"])

df_filled['hour'] = df_filled['datetime'].dt.hour.astype(int)
df_filled['day'] = df_filled['datetime'].dt.day.astype(int)
df_filled['month'] = df_filled['datetime'].dt.month.astype(int)
df_filled['year'] = df_filled['datetime'].dt.year.astype(int)
df_filled.head()
```

Out[136]:
```
             datetime                   city state country   latitude  \
0 1949-10-10 20:30:00             san marcos    tx      us  29.883056
1 1949-10-10 21:00:00           lackland afb    tx      us  29.384210
2 1955-10-10 17:00:00   chester (uk/england)   NaN      gb  53.200000
```

```
          3 1956-10-10 21:00:00                  edna    tx      us  28.978333
          4 1960-10-10 20:00:00               kaneohe    hi      us  21.418056

              longitude  duration  hour  day  month  year
          0  -97.941111    2700.0    20   10     10  1949
          1  -98.581082    7200.0    21   10     10  1949
          2   -2.916667      20.0    17   10     10  1955
          3  -96.645833      20.0    21   10     10  1956
          4 -157.803611     900.0    20   10     10  1960
```

```
In [137]: df_filled.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 87458 entries, 0 to 88678
Data columns (total 11 columns):
datetime     87458 non-null datetime64[ns]
city         87458 non-null object
state        80269 non-null object
country      82954 non-null object
latitude     87458 non-null float64
longitude    87458 non-null float64
duration     87458 non-null float32
hour         87458 non-null int64
day          87458 non-null int64
month        87458 non-null int64
year         87458 non-null int64
dtypes: datetime64[ns](1), float32(1), float64(2), int64(4), object(3)
memory usage: 7.7+ MB
```

```
In [140]: df_filled.to_csv("../data/cleaned.csv", index=False)
          comments.to_csv("../data/comments.csv", index=False)
          shapes.to_csv("../data/shapes.csv", index=False)
```