

# population

April 14, 2023

```
[112]: import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import statsmodels.api as sm
import scipy.stats as st
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import train_test_split
from datetime import date, datetime

%matplotlib inline
```

```
[113]: df_out = pd.read_pickle('df_out.pkl')
df_out_with_breeds_info = pd.read_pickle('df_out_with_breeds_info.pkl')
df_out.info()
df_out_with_breeds_info.info()
df_out.head()
```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 149511 entries, 0 to 149510

Data columns (total 40 columns):

| #  | Column            | Non-Null Count  | Dtype          |
|----|-------------------|-----------------|----------------|
| 0  | Animal ID         | 149511 non-null | string         |
| 1  | Name              | 106260 non-null | string         |
| 2  | Outcome DateTime  | 149511 non-null | datetime64[ns] |
| 3  | Outcome MonthYear | 149511 non-null | string         |
| 4  | Date of Birth     | 149511 non-null | datetime64[ns] |
| 5  | Outcome Type      | 149485 non-null | string         |
| 6  | Outcome Subtype   | 68443 non-null  | string         |
| 7  | Animal Type       | 149511 non-null | string         |
| 8  | Sex upon Outcome  | 149509 non-null | string         |
| 9  | Age upon Outcome  | 149465 non-null | string         |
| 10 | Breed             | 149511 non-null | string         |
| 11 | Color             | 149511 non-null | string         |
| 12 | Intake MonthYear  | 136236 non-null | string         |

|    |                          |        |          |                |
|----|--------------------------|--------|----------|----------------|
| 13 | Intake DateTime          | 136236 | non-null | datetime64[ns] |
| 14 | Found Location           | 136236 | non-null | string         |
| 15 | Intake Type              | 136236 | non-null | string         |
| 16 | Intake Condition         | 136236 | non-null | string         |
| 17 | Sex upon Intake          | 136234 | non-null | string         |
| 18 | Age upon Intake          | 136235 | non-null | string         |
| 19 | Years in animal center   | 136236 | non-null | Float64        |
| 20 | Colors (count)           | 149511 | non-null | Int64          |
| 21 | Color 0                  | 149511 | non-null | string         |
| 22 | Color 1                  | 79869  | non-null | string         |
| 23 | Color 0 R                | 135638 | non-null | Float64        |
| 24 | Color 0 G                | 135638 | non-null | Float64        |
| 25 | Color 0 B                | 135638 | non-null | Float64        |
| 26 | Color 0 H                | 135638 | non-null | Float64        |
| 27 | Color 0 S                | 135638 | non-null | Float64        |
| 28 | Color 0 V                | 135638 | non-null | Float64        |
| 29 | Color 1 R                | 78596  | non-null | Float64        |
| 30 | Color 1 G                | 78596  | non-null | Float64        |
| 31 | Color 1 B                | 78596  | non-null | Float64        |
| 32 | Color 1 H                | 78596  | non-null | Float64        |
| 33 | Color 1 S                | 78596  | non-null | Float64        |
| 34 | Color 1 V                | 78596  | non-null | Float64        |
| 35 | Age upon Outcome (years) | 149465 | non-null | Float64        |
| 36 | Male                     | 149509 | non-null | boolean        |
| 37 | Female                   | 149509 | non-null | boolean        |
| 38 | NeuteredOrSpayed         | 149509 | non-null | boolean        |
| 39 | Adopted                  | 149485 | non-null | boolean        |

dtypes: Float64(14), Int64(1), boolean(4), datetime64[ns](3), string(18)

memory usage: 44.3 MB

<class 'pandas.core.frame.DataFrame'>

Int64Index: 149511 entries, 0 to 149510

Data columns (total 56 columns):

| #  | Column            | Non-Null Count  | Dtype          |
|----|-------------------|-----------------|----------------|
| 0  | Animal ID         | 149511 non-null | string         |
| 1  | Name              | 106260 non-null | string         |
| 2  | Outcome DateTime  | 149511 non-null | datetime64[ns] |
| 3  | Outcome MonthYear | 149511 non-null | string         |
| 4  | Date of Birth     | 149511 non-null | datetime64[ns] |
| 5  | Outcome Type      | 149485 non-null | string         |
| 6  | Outcome Subtype   | 68443 non-null  | string         |
| 7  | Animal Type       | 149511 non-null | string         |
| 8  | Sex upon Outcome  | 149509 non-null | string         |
| 9  | Age upon Outcome  | 149465 non-null | string         |
| 10 | Breed             | 149511 non-null | string         |
| 11 | Color             | 149511 non-null | string         |
| 12 | Intake MonthYear  | 136236 non-null | string         |
| 13 | Intake DateTime   | 136236 non-null | datetime64[ns] |

|    |                          |        |          |         |
|----|--------------------------|--------|----------|---------|
| 14 | Found Location           | 136236 | non-null | string  |
| 15 | Intake Type              | 136236 | non-null | string  |
| 16 | Intake Condition         | 136236 | non-null | string  |
| 17 | Sex upon Intake          | 136234 | non-null | string  |
| 18 | Age upon Intake          | 136235 | non-null | string  |
| 19 | Years in animal center   | 136236 | non-null | Float64 |
| 20 | Colors (count)           | 149511 | non-null | Int64   |
| 21 | Color 0                  | 149511 | non-null | string  |
| 22 | Color 1                  | 79869  | non-null | string  |
| 23 | Color 0 R                | 135638 | non-null | Float64 |
| 24 | Color 0 G                | 135638 | non-null | Float64 |
| 25 | Color 0 B                | 135638 | non-null | Float64 |
| 26 | Color 0 H                | 135638 | non-null | Float64 |
| 27 | Color 0 S                | 135638 | non-null | Float64 |
| 28 | Color 0 V                | 135638 | non-null | Float64 |
| 29 | Color 1 R                | 78596  | non-null | Float64 |
| 30 | Color 1 G                | 78596  | non-null | Float64 |
| 31 | Color 1 B                | 78596  | non-null | Float64 |
| 32 | Color 1 H                | 78596  | non-null | Float64 |
| 33 | Color 1 S                | 78596  | non-null | Float64 |
| 34 | Color 1 V                | 78596  | non-null | Float64 |
| 35 | Age upon Outcome (years) | 149465 | non-null | Float64 |
| 36 | Male                     | 149509 | non-null | boolean |
| 37 | Female                   | 149509 | non-null | boolean |
| 38 | NeuteredOrSpayed         | 149509 | non-null | boolean |
| 39 | Adopted                  | 149485 | non-null | boolean |
| 40 | BreedsInfoName           | 138419 | non-null | object  |
| 41 | Breed (catalog)          | 138419 | non-null | string  |
| 42 | Breed Group AKC          | 138419 | non-null | string  |
| 43 | Breed Group CKC          | 138419 | non-null | string  |
| 44 | Breed Group UKC          | 138419 | non-null | string  |
| 45 | CKC Subgroup             | 138414 | non-null | string  |
| 46 | height_low_inches        | 138419 | non-null | Float64 |
| 47 | height_high_inches       | 138419 | non-null | Float64 |
| 48 | average height           | 138419 | non-null | Float64 |
| 49 | weight_low_lbs           | 138419 | non-null | Float64 |
| 50 | weight_high_lbs          | 138419 | non-null | Int64   |
| 51 | average weight           | 138419 | non-null | Float64 |
| 52 | Lifespan Low             | 138415 | non-null | Int64   |
| 53 | Lifespan High            | 138415 | non-null | Int64   |
| 54 | average lifespan         | 138419 | non-null | Float64 |
| 55 | Est. lifespan remaining  | 138408 | non-null | float64 |

dtypes: Float64(20), Int64(4), boolean(4), datetime64[ns](3), float64(1), object(1), string(23)

memory usage: 65.0+ MB

```
[113]: Animal ID      Name      Outcome DateTime Outcome MonthYear Date of Birth \
0      A794011  Chunk  2019-05-08 18:20:00          May 2019   2017-05-02
1      A776359  Gizmo  2018-07-18 16:02:00          Jul 2018   2017-07-12
2      A821648  <NA>  2020-08-16 11:38:00          Aug 2020   2019-08-16
3      A720371  Moose  2016-02-13 17:59:00          Feb 2016   2015-10-08
4      A674754  <NA>  2014-03-18 11:47:00          Mar 2014   2014-03-12

Outcome Type Outcome Subtype Animal Type Sex upon Outcome Age upon Outcome \
0      Rto-Adopt          <NA>      Cat      Neutered Male          2 years
1      Adoption          <NA>      Dog      Neutered Male          1 year
2      Euthanasia        <NA>      Other      Unknown          1 year
3      Adoption          <NA>      Dog      Neutered Male          4 months
4      Transfer          Partner      Cat      Intact Male          6 days

... Color 1 G Color 1 B Color 1 H Color 1 S Color 1 V \
0      ...          1.0          1.0          0.0          0.0          1.0
1      ...          0.44          0.09  0.119444          0.85          0.59
2      ...          <NA>          <NA>          <NA>          <NA>          <NA>
3      ...          <NA>          <NA>          <NA>          <NA>          <NA>
4      ...          <NA>          <NA>          <NA>          <NA>          <NA>

Age upon Outcome (years)      Male Female NeuteredOrSpayed      Adopted
0              2.0      True  False          True      True
1              1.0      True  False          True      True
2              1.0     False  False          False     False
3              0.333333     True  False          True      True
4              0.016438     True  False          False     False
```

[5 rows x 40 columns]

```
[114]: def population(start, end):
        loc_intake = (df_out['Intake DateTime'].isna() | (df_out['Intake DateTime'] >
        < end))
        loc_outcome = (df_out['Outcome DateTime'].isna() | (df_out['Outcome
        < DateTime'] >= start))
        return df_out.loc[loc_intake & loc_outcome]
```

```
[115]: start = datetime(2014, 1, 1)
end = datetime(2023, 4, 1)
offset = datetime(start.year, start.month, start.day)

window_start = 'Window start'
window_end = 'Window end'
feature_1 = 'Intakes'
feature_1x_classes = df_out['Intake Type'].unique()
feature_1y_classes = df_out['Intake Condition'].unique()
```

```

feature_1x = [f'Intakes Types ({feature_class})' for feature_class in
    ↪feature_1x_classes]
feature_1y = [f'Intakes Conditions ({feature_class})' for feature_class in
    ↪feature_1y_classes]
feature_2 = 'Outcomes'
feature_2x_classes = df_out['Outcome Type'].unique()
feature_2x = [f'Outcomes ({feature_class})' for feature_class in
    ↪feature_2x_classes]
feature_3 = 'Breeds (unique)'
feature_4 = 'Animals (count)'

df_populations = pd.DataFrame()

df_populations[window_start] = pd.Series(dtype=df_out["Outcome DateTime"].dtype)
df_populations[window_end] = pd.Series(dtype=df_out["Outcome DateTime"].dtype)

df_populations.set_index(window_start)

df_populations[feature_1] = pd.Series(dtype=int)
for feature in feature_1x:
    df_populations[f'{feature} (Absolute)'] = pd.Series(dtype=int)
    df_populations[f'{feature} (Relative)'] = pd.Series(dtype=float)
df_populations = df_populations.copy()
for feature in feature_1y:
    df_populations[feature] = pd.Series(dtype=int)
    df_populations[f'{feature} (Absolute)'] = pd.Series(dtype=int)
    df_populations[f'{feature} (Relative)'] = pd.Series(dtype=float)
df_populations = df_populations.copy()

df_populations[feature_2] = pd.Series(dtype=int)
for feature in feature_2x:
    df_populations[feature] = pd.Series(dtype=int)
    df_populations[f'{feature} (Absolute)'] = pd.Series(dtype=int)
    df_populations[f'{feature} (Relative)'] = pd.Series(dtype=float)
df_populations = df_populations.copy()

df_populations[feature_3] = pd.Series(dtype=int)

df_populations = df_populations.copy()

while offset != end:
    offset_next = datetime(offset.year + (1 if offset.month == 12 else 0),
    ↪(offset.month % 12) + 1, offset.day)

    df_populations.at[offset, window_start] = offset
    df_populations.at[offset, window_end] = offset_next

```

```

animals = population(offset, offset_next)

intakes = animals.loc[(animals['Intake DateTime'] >= offset) &
↳(animals['Intake DateTime'] < offset_next)]
df_populations.at[offset, feature_1] = intakes.shape[0]
for feature_class, count in intakes['Intake Type'].value_counts().items():
    df_populations.at[offset, f'Intakes Types ({feature_class})']
↳(Absolute)'] = count
    df_populations.at[offset, f'Intakes Types ({feature_class})']
↳(Relative)'] = count / intakes.shape[0]
    for feature_class, count in intakes['Intake Condition'].value_counts().
↳items():
        df_populations.at[offset, f'Intakes Conditions ({feature_class})']
↳(Absolute)'] = count
        df_populations.at[offset, f'Intakes Conditions ({feature_class})']
↳(Relative)'] = count / intakes.shape[0]

outcomes = animals.loc[(animals['Outcome DateTime'] >= offset) &
↳(animals['Outcome DateTime'] < offset_next)]
df_populations.at[offset, feature_2] = outcomes.shape[0]
for feature_class, count in outcomes['Outcome Type'].value_counts().items():
    df_populations.at[offset, f'Outcome Types ({feature_class})']
↳(Absolute)'] = count
    df_populations.at[offset, f'Outcome Types ({feature_class})']
↳(Relative)'] = count / outcomes.shape[0]

df_populations.at[offset, feature_3] = len(animals['Breed'].unique())
df_populations.at[offset, feature_4] = animals.shape[0]

offset = offset_next

```

```

[116]: df_populations['Month'] = pd.Series(dtype=int)
df_populations['Year'] = pd.Series(dtype=int)
for index in df_populations.index:
    df_populations.at[index, 'Month'] = df_populations.at[index, window_start].
↳month
    df_populations.at[index, 'Year'] = df_populations.at[index, window_start].
↳year

```

```

[117]: def populationCharts(feature):
    plt.figure(figsize=(12, 5))
    sns.lineplot(
        data=df_populations,
        x=window_start,
        y=feature
    )

```

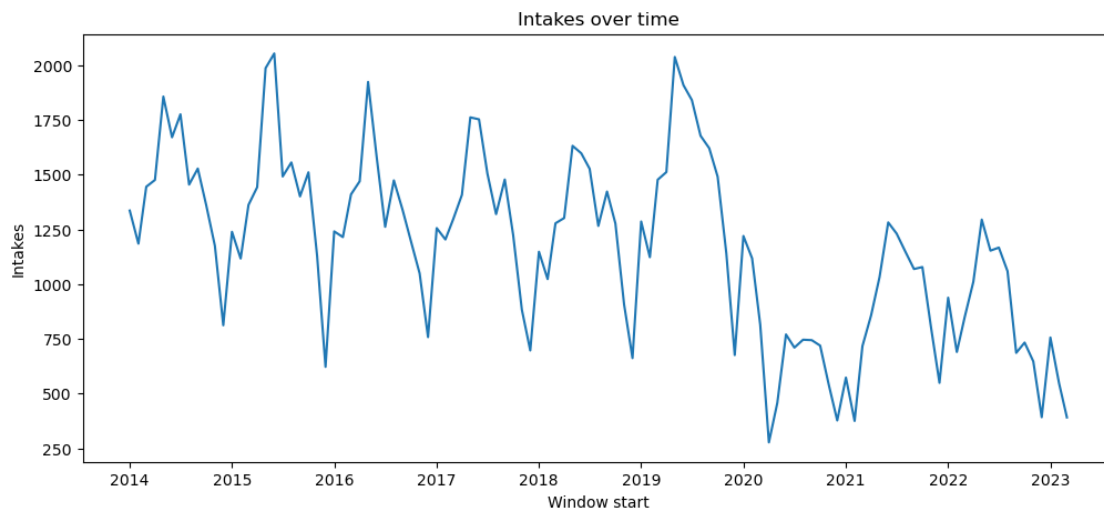
```

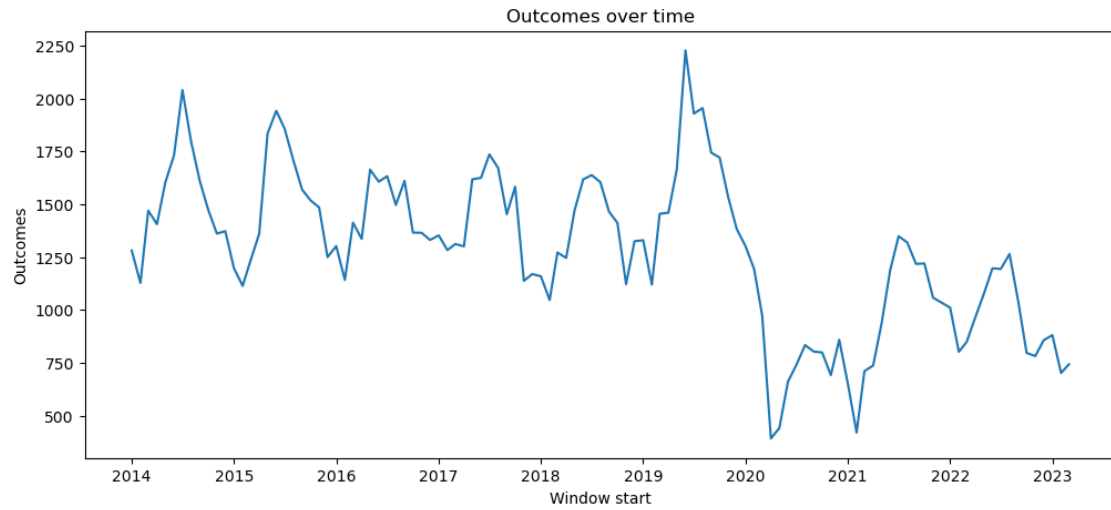
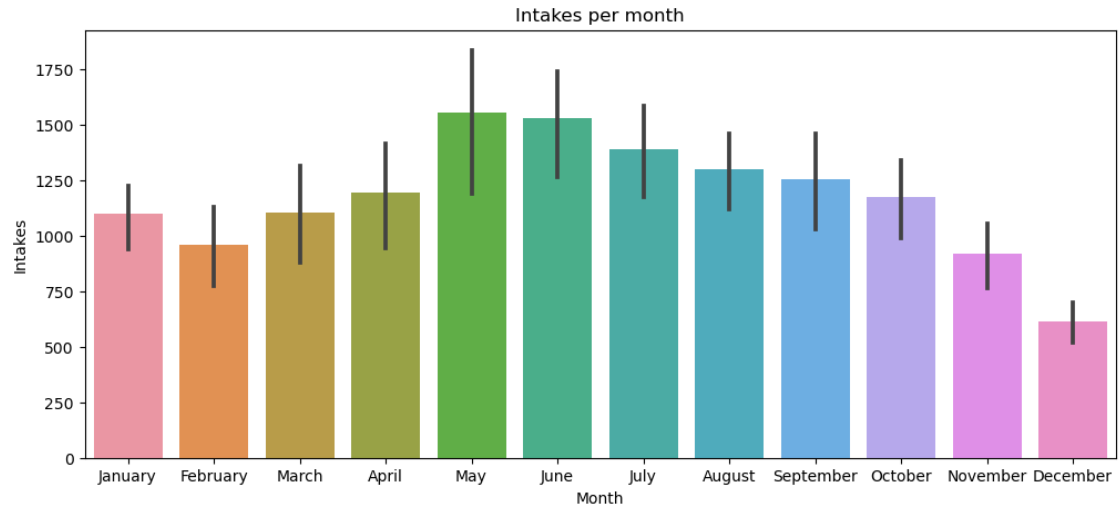
plt.title(f'{feature} over time')
plt.show()

plt.figure(figsize=(12, 5))
sns.barplot(
    data=df_populations,
    x='Month',
    y=feature,
)
plt.xticks(range(12), ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"])
plt.title(f'{feature} per month')
plt.show()

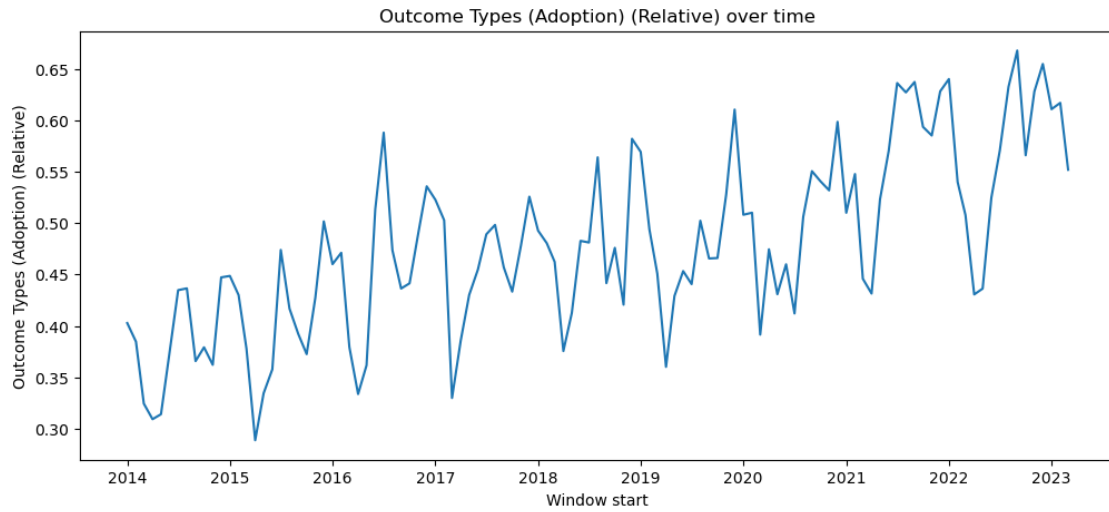
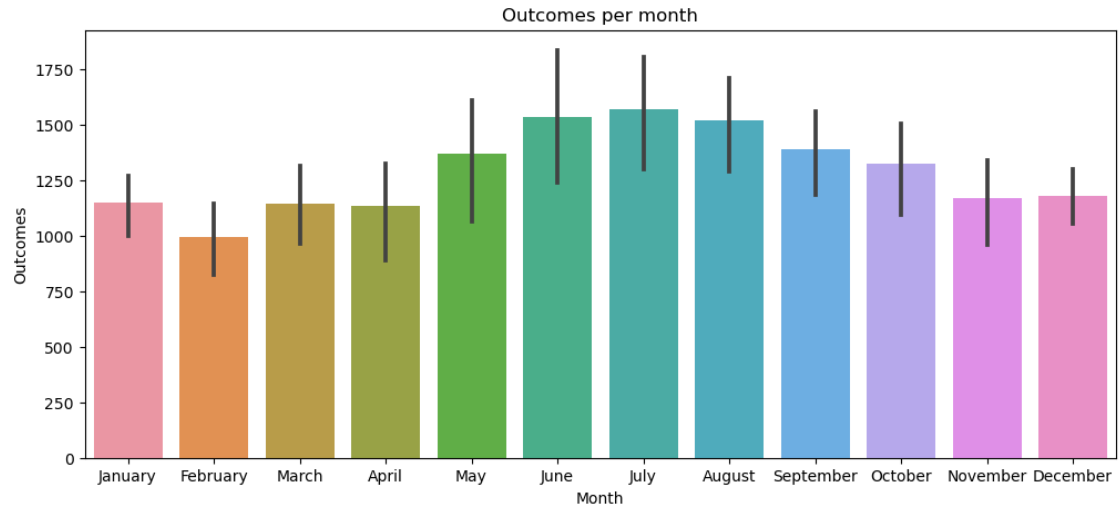
populationCharts('Intakes')
populationCharts('Outcomes')
populationCharts('Outcome Types (Adoption) (Relative)')

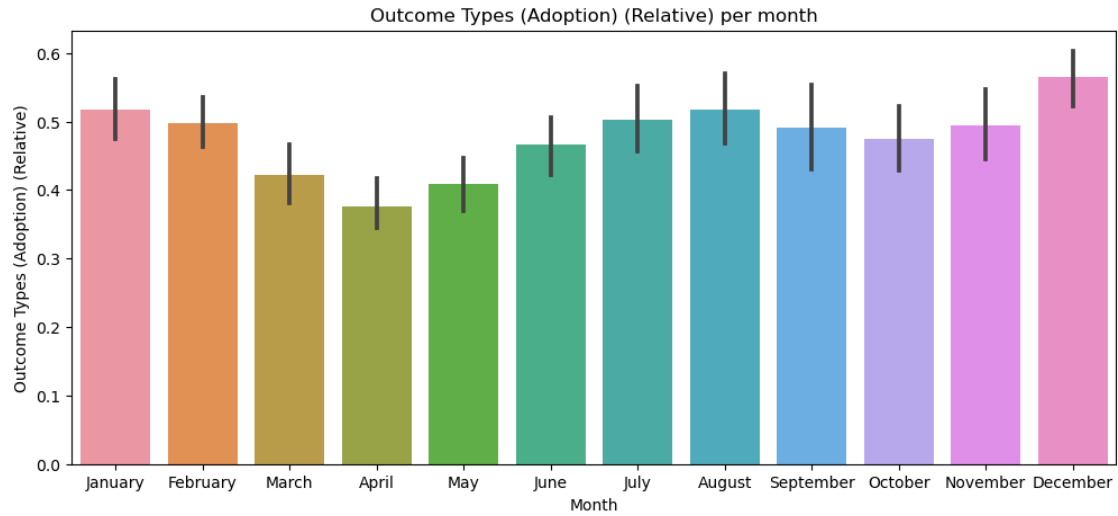
```











```
[118]: df_populations.Year.head(20)
```

```
[118]: 2014-01-01    2014.0
2014-02-01    2014.0
2014-03-01    2014.0
2014-04-01    2014.0
2014-05-01    2014.0
2014-06-01    2014.0
2014-07-01    2014.0
2014-08-01    2014.0
2014-09-01    2014.0
2014-10-01    2014.0
2014-11-01    2014.0
2014-12-01    2014.0
2015-01-01    2015.0
2015-02-01    2015.0
2015-03-01    2015.0
2015-04-01    2015.0
2015-05-01    2015.0
2015-06-01    2015.0
2015-07-01    2015.0
2015-08-01    2015.0
Name: Year, dtype: float64
```

```
[119]: def adoptionCorr(feature, hue):
sns.scatterplot(
    data=df_populations,
    x=feature,
    y='Outcome Types (Adoption) (Relative)',
```

```

        hue=hue
    )
    plt.title(f"Relationship between {feature} and likelihood of adoption")
    plt.show()

    try:
        sns.jointplot(
            data=df_populations,
            x=feature,
            y='Outcome Types (Adoption) (Relative)',
            kind='reg'
        )
        plt.show()
    except: pass

adoptionCorr(feature_3, 'Year')
adoptionCorr(feature_4, 'Year')
adoptionCorr(window_start, None)

```

