eda

February 27, 2026

1 Central Park Squirrels Exploratory Data Analysis

1.1 Imports

[1]:
```
import pandas as pd
import altair as alt
alt.data_transformers.enable('vegafusion')
```

[1]:
```
DataTransformerRegistry.enable('vegafusion')
```

1.2 Loading & Cleaning

[ ]:
```
squirrels = pd.read_csv('../data/raw/2018_Central_Park_Squirrel_Census.csv')

squirrels['Date'] = pd.to_datetime(squirrels['Date'], format='%m%d%Y')
squirrels.columns = squirrels.columns.str.lower().str.replace(' ', '_')

squirrels.to_csv('../data/processed/squirrels.csv', index=False)
```

[ ]:
```
Index(['x', 'y', 'unique_squirrel_id', 'hectare', 'shift', 'date',
       'hectare_squirrel_number', 'age', 'primary_fur_color',
       'highlight_fur_color', 'combination_of_primary_and_highlight_color',
       'color_notes', 'location', 'above_ground_sighter_measurement',
       'specific_location', 'running', 'chasing', 'climbing', 'eating',
       'foraging', 'other_activities', 'kuks', 'quaas', 'moans', 'tail_flags',
       'tail_twitches', 'approaches', 'indifferent', 'runs_from',
       'other_interactions', 'lat/long'],
      dtype='str')
```

1.3 Dataset Overview

[ ]:
```
squirrels.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3023 entries, 0 to 3022
Data columns (total 31 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   x                                           3023 non-null   float64
 1   y                                           3023 non-null   float64
 2   unique_squirrel_id                          3023 non-null   object
 3   hectare                                     3023 non-null   object
 4   shift                                       3023 non-null   object
 5   date                                        3023 non-null   datetime64[ns]
 6   hectare_squirrel_number                     3023 non-null   int64
 7   age                                         2902 non-null   object
 8   primary_fur_color                           2968 non-null   object
 9   highlight_fur_color                         1937 non-null   object
 10  combination_of_primary_and_highlight_color  3023 non-null   object
 11  color_notes                                 182 non-null    object
 12  location                                    2959 non-null   object
 13  above_ground_sighter_measurement            2909 non-null   object
 14  specific_location                           476 non-null    object
 15  running                                     3023 non-null   bool
 16  chasing                                     3023 non-null   bool
 17  climbing                                    3023 non-null   bool
 18  eating                                      3023 non-null   bool
 19  foraging                                    3023 non-null   bool
 20  other_activities                            437 non-null    object
 21  kuks                                        3023 non-null   bool
 22  quaas                                       3023 non-null   bool
 23  moans                                       3023 non-null   bool
 24  tail_flags                                  3023 non-null   bool
 25  tail_twitches                               3023 non-null   bool
 26  approaches                                  3023 non-null   bool
 27  indifferent                                 3023 non-null   bool
 28  runs_from                                   3023 non-null   bool
 29  other_interactions                          240 non-null    object
 30  lat/long                                    3023 non-null   object
dtypes: bool(13), datetime64[ns](1), float64(2), int64(1), object(14)
memory usage: 463.6+ KB
```

1.4 EDA

[ ]:

```
colours = ['#B2BEB5', '#D2691E', '#000000']
order = ['Gray', 'Cinnamon', 'Black']

colour_sightings = alt.Chart(
    squirrels.dropna(subset = ['primary_fur_color']),
    title = alt.Title(text = 'Most Common Fur Colours')
    ).mark_bar().encode(
        x = alt.X('primary_fur_color:N').sort('-y').title('Primary Fur Colour'),
        y = alt.Y('count():Q').title('Number of Sightings'),
        color = alt.Color('primary_fur_color:N').scale(domain = order, range =␣
 ↪colours).legend(None)
    ).properties(width = 400, height = 250)
```

```
colour_sightings
```

[ ]: alt.Chart(…)

[ ]:

```
location_sightings = alt.Chart(
    squirrels.dropna(subset = ['x', 'y']),
    title = alt.Title(text = 'Squirrel Sightings by Location and Colour')
    ).mark_circle(size = 15, opacity = 0.5).encode(
        x = alt.X('x:Q').title('Longitude (x)').scale(zero = False),
        y = alt.Y('y:Q').title('Latitude (y)').scale(zero = False),
        color = alt.Color('primary_fur_color:N').scale(domain = order, range =␣
 ↪colours).title('Primary Fur Colour'),
    ).properties(width = 400, height = 250)

location_sightings
```

[ ]: alt.Chart(…)

[ ]:

```
day_night_sightings = alt.Chart(
    squirrels.dropna(subset = ['primary_fur_color', 'shift'])
    ).mark_bar().encode(
        x = alt.X('primary_fur_color:N').sort('-y').title(None),
        y = alt.Y('count():Q').title('Number of Sightings'),
        color = alt.Color('primary_fur_color:N').scale(domain = order, range =␣
 ↪colours).legend(None)
    ).facet(
        column = alt.Column('shift:N', title = 'Time of Day')
    )

day_night_sightings
```

[ ]: alt.FacetChart(…)

[ ]:

```
cumulative_sightings = alt.Chart(
    squirrels.dropna(subset = ['primary_fur_color', 'date']),
    title = alt.Title(text = 'Cumulative Sightings of Different Fur Colours␣
 ↪Over Time')
    ).transform_aggregate(
        count = 'count()',
        groupby = ['date', 'primary_fur_color']
    ).transform_window(
        cumulative_count = 'sum(count)',
        sort = [alt.SortField('date')],
        groupby = ['primary_fur_color']
    ).mark_line().encode(
        x = alt.X('date:T', title = 'Date'),
        y = alt.Y('cumulative_count:Q', title = 'Cumulative Sightings'),
        color = alt.Color('primary_fur_color:N').scale(domain = order, range =␣
  ↪colours).title('Primary Fur Colour')
    ).properties(width = 400, height = 250)

cumulative_sightings
```

[ ]: alt.Chart(…)

[ ]:
```
vocal_cols = ['kuks', 'quaas', 'moans']

squirrel_vocals = (
    squirrels[['primary_fur_color'] + vocal_cols]
    .dropna(subset = ['primary_fur_color'])
    .assign(
        any_vocal = lambda df: df[vocal_cols]
            .fillna(False)
            .astype(bool)
            .any(axis=1)
    )
    [['primary_fur_color', 'any_vocal']]
)

colour_noise = alt.Chart(
    squirrel_vocals,
    title = alt.Title(text = 'Proportion of Squirrels Making Vocalisations by␣
  ↪Fur Colour')
).mark_bar().encode(
    x = alt.X('primary_fur_color:N').sort('-y').title('Primary Fur Colour'),
    y = alt.Y('mean(any_vocal):Q', title = 'Proportion of Squirrels Making␣
  ↪Vocalisations'),
    color = alt.Color('primary_fur_color:N').scale(domain = order, range =␣
  ↪colours).legend(None)
).properties(width = 400, height = 250)

colour_noise
```

[ ]: alt.Chart(…)

[ ]:
```
colour_run = alt.Chart(
    squirrels.dropna(subset = ['primary_fur_color', 'runs_from']),
    title = alt.Title(text = 'Proportion of Squirrels That Run From Humans by␣
  ↪Fur Colour')
    ).mark_bar().encode(
        x = alt.X('primary_fur_color:N').sort('-y').title('Primary Fur Colour'),
        y = alt.Y('mean(runs_from):Q', title = 'Proportion of Squirrels That␣
  ↪Run From Humans'),
        color = alt.Color('primary_fur_color:N').scale(domain = order, range = colours).legend(None)
    ).properties(width = 400, height = 250)

colour_run
```

[ ]: alt.Chart(…)

[ ]:
```
colour_eat = alt.Chart(
    squirrels.dropna(subset = ['primary_fur_color', 'eating']),
    title = alt.Title(text = 'Proportion of Squirrels Eating When Sighted by Fur Colour')
    ).mark_bar().encode(
        x = alt.X('primary_fur_color:N').sort('-y').title('Primary Fur Colour'),
        y = alt.Y('mean(eating):Q', title = 'Proportion of Squirrels That Are Eating'),
        color = alt.Color('primary_fur_color:N').scale(domain = order, range = colours).legend(None)
    ).properties(width = 400, height = 250)

colour_eat
```

[ ]: alt.Chart(…)