02-understanding-dataset

March 24, 2025

1 Understanding Dataset Exercise

Import necessary libraries

[25]:
```
import pandas as pd
import os
import opendatasets as od
import matplotlib.pyplot as plt
import seaborn as sns
```

Download and save dataset from Kaggle

[ ]:
```
# I'm using sloth data

url = "https://www.kaggle.com/datasets/bertiemackie/sloth-species?
 ↪select=sloth_data.csv"
target_dir = target_folder = "../../data"

od.download(url, data_dir=target_folder)
```

```
Please provide your Kaggle credentials to download this dataset. Learn more:
http://bit.ly/kaggle-creds
Your Kaggle username:Your Kaggle username:
```

[2]:
```
df = pd.read_csv("../../data/sloth.csv")
```

Display the dataset

[3]:
```
df.head()
```

[3]:

	Unnamed: 0	claw_length_cm	endangered	size_cm	specie	sub_specie	tail_length_cm	weight_kg
0	0	6.825	critically_endangered	52.004	three_toed	Pygmy three-toed sloth	4.448	3.570
1	1	8.260	critically_endangered	50.082	three_toed	Pygmy three-toed sloth	6.286	2.844
2	2	8.662	critically_endangered	51.498	three_toed			
3	3	8.467	critically_endangered	50.122	three_toed			
4	4	7.104	critically_endangered	51.364	three_toed			

```
2  Pygmy three-toed sloth          4.551      1.259
3  Pygmy three-toed sloth          6.983      2.392
4  Pygmy three-toed sloth          5.411      3.163
```

Check columns names

[4]:
```
df.columns
```

[4]:
```
Index(['Unnamed: 0', 'claw_length_cm', 'endangered', 'size_cm', 'specie',
       'sub_specie', 'tail_length_cm', 'weight_kg'],
      dtype='object')
```

Check missing values

[5]:
```
df.isna().any()
```

[5]:
```
Unnamed: 0        False
claw_length_cm    False
endangered        False
size_cm           False
specie            False
sub_specie        False
tail_length_cm    False
weight_kg         False
dtype: bool
```

Check data types

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Unnamed: 0      5000 non-null   int64
 1   claw_length_cm  5000 non-null   float64
 2   endangered      5000 non-null   object
 3   size_cm         5000 non-null   float64
 4   specie          5000 non-null   object
 5   sub_specie      5000 non-null   object
 6   tail_length_cm  5000 non-null   float64
 7   weight_kg       5000 non-null   float64
dtypes: float64(4), int64(1), object(3)
memory usage: 312.6+ KB
```

Produce simple statistic

[7]:
```
df.describe()
```

[7]:

	Unnamed: 0	claw_length_cm	size_cm	tail_length_cm	weight_kg
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000
mean	2499.500000	7.423503	60.399852	3.410544	5.253253
std	1443.520003	1.520533	5.929968	2.333288	1.268203
min	0.000000	1.748000	46.928000	-2.942000	0.946000
25%	1249.750000	6.383750	59.904750	1.440250	4.382500
50%	2499.500000	7.445000	62.478500	3.812000	5.274000
75%	3749.250000	8.491500	64.398250	5.351250	6.125250
max	4999.000000	12.171000	68.760000	8.538000	9.997000

Generate simple plot

You should generate at least 5 charts

You can refer to this websites:

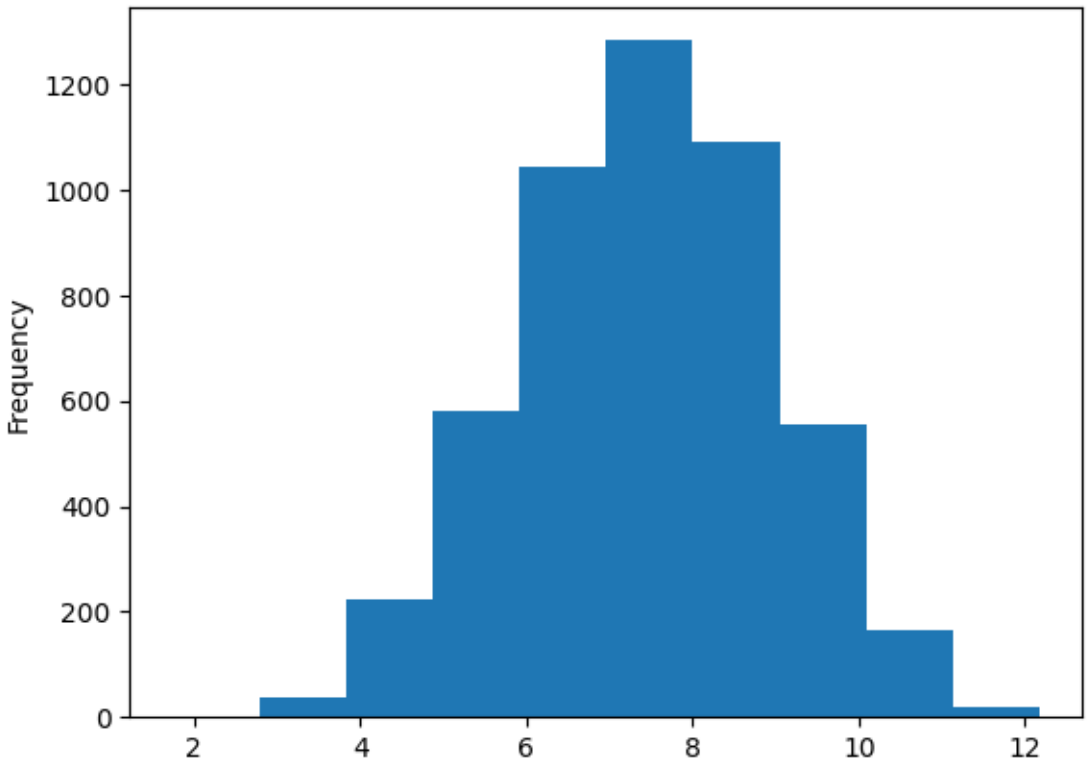
https://www.w3schools.com/python/pandas/pandas_plotting.asp

https://pandas.pydata.org/docs/user_guide/visualization.html

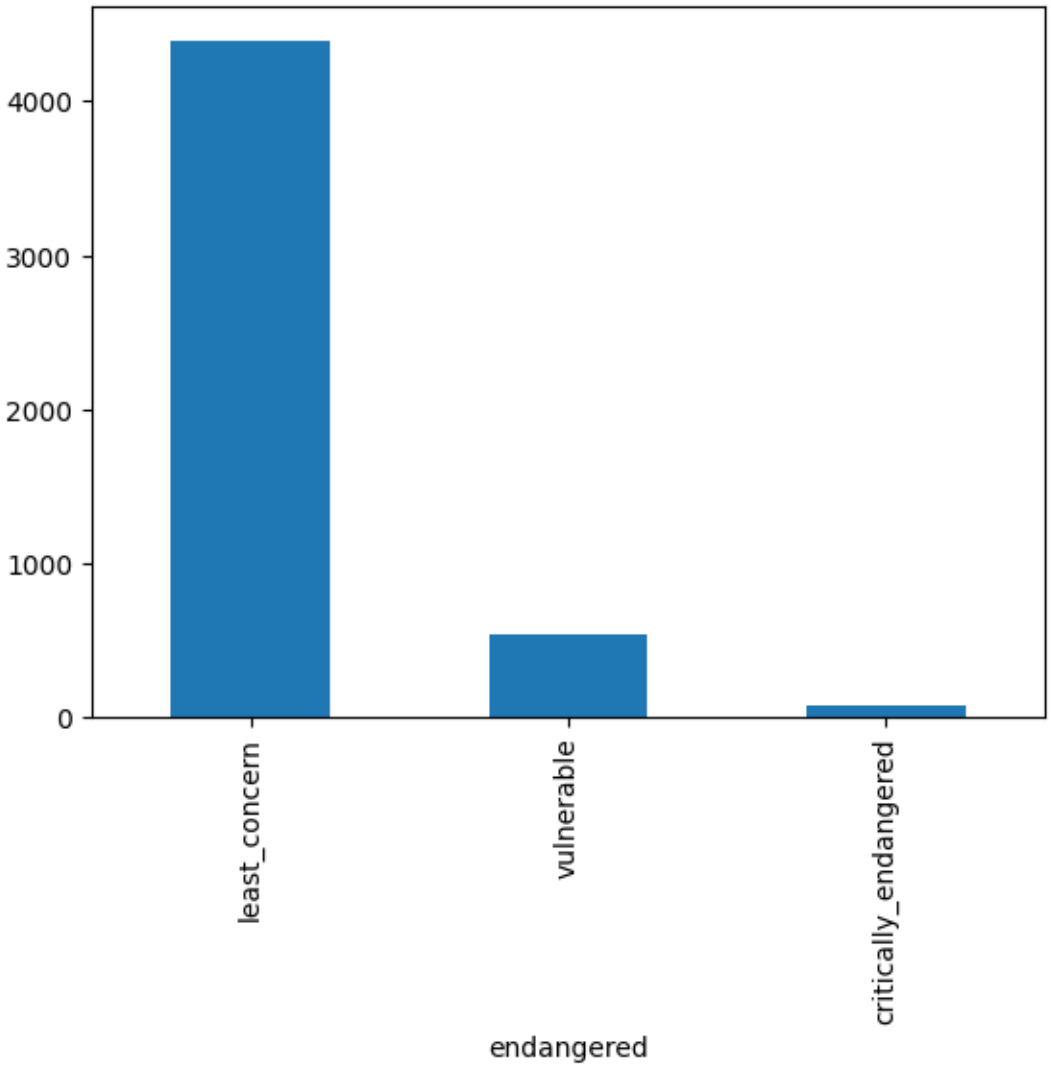
or just google search to find any interesting plot diagram using any of these keywords:

'python chart, pandas chart, matplotlib'

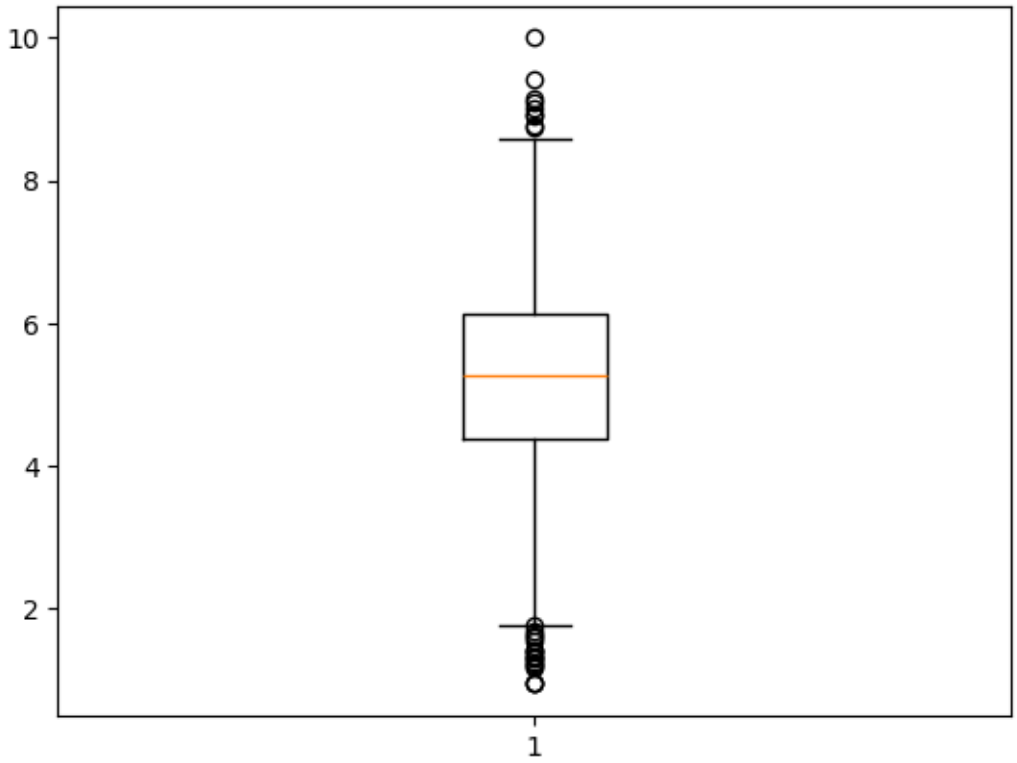
[18]:
```
df["claw_length_cm"].plot.hist();
```



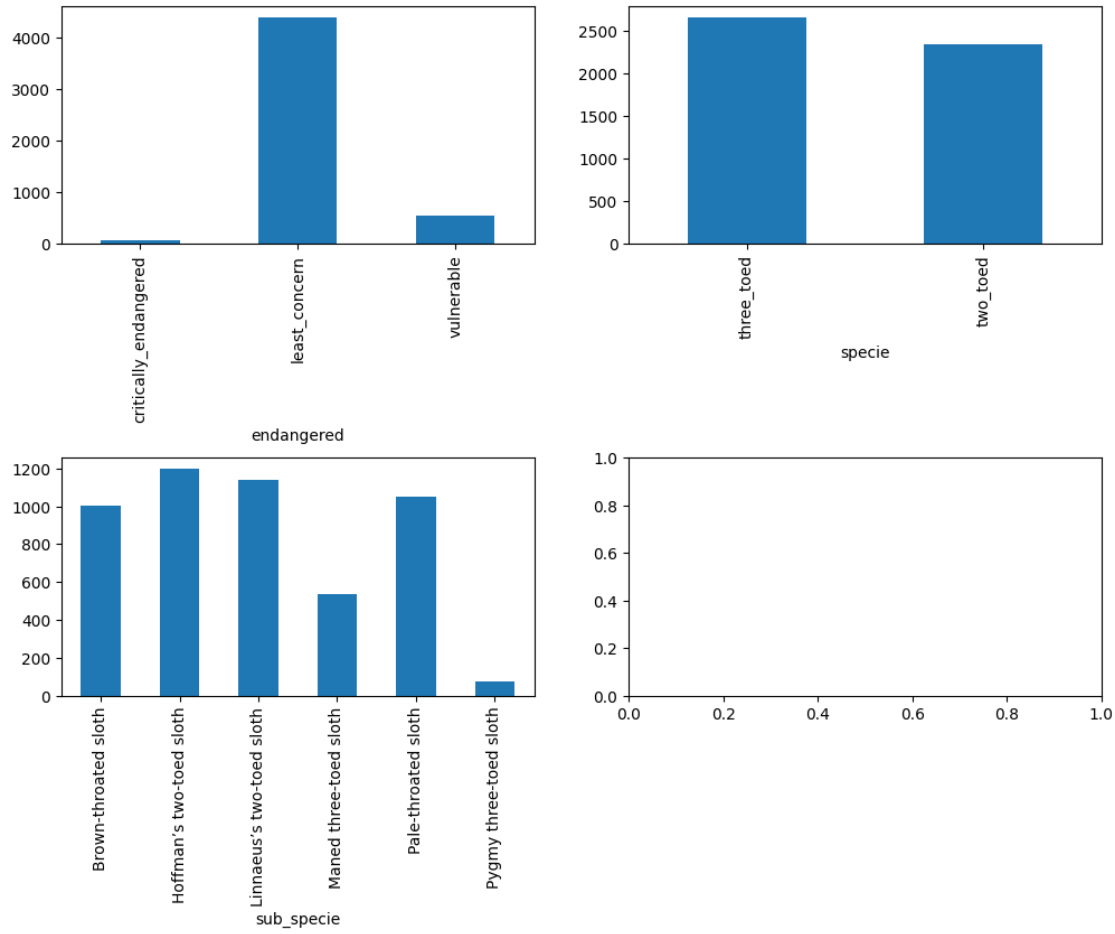
[17]:
```
df["endangered"].value_counts().plot.bar();
```



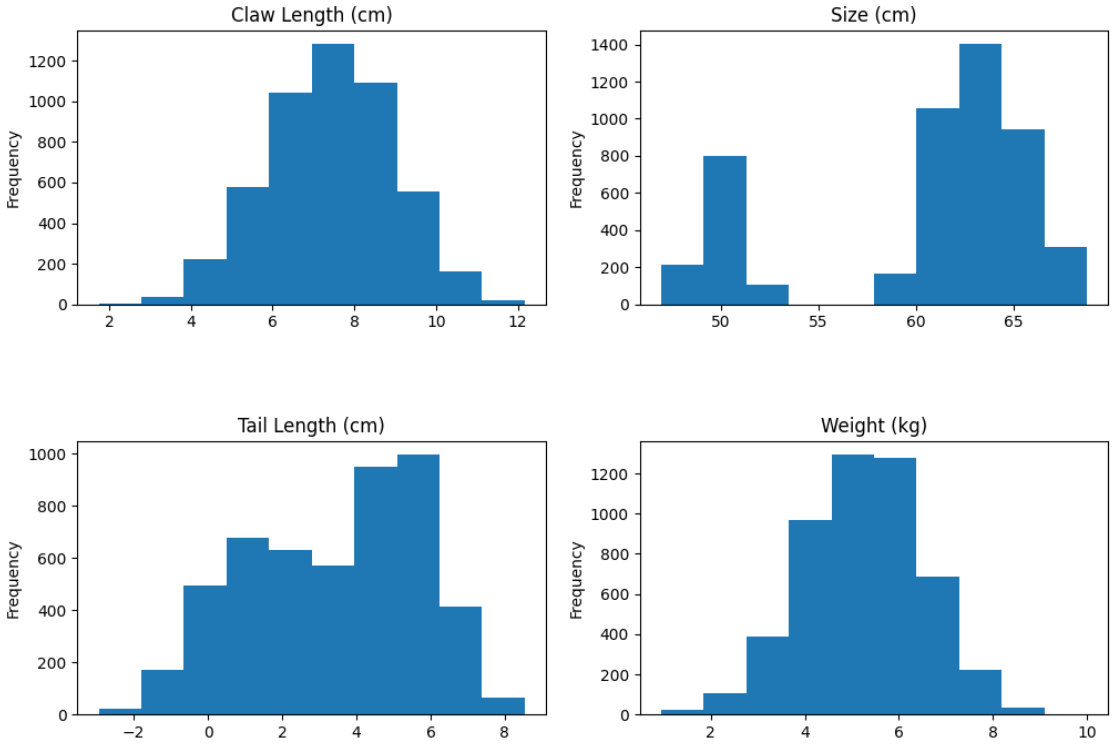
[22]:
```
plt.boxplot(df["weight_kg"]);
```



[ ]:
```
fig, ax = plt.subplots(2,2, figsize=(12,8))

df["endangered"].value_counts().sort_index().plot.bar(ax=ax[0][0])
df["specie"].value_counts().sort_index().plot.bar(ax=ax[0][1])
df["sub_specie"].value_counts().sort_index().plot.bar(ax=ax[1][0])

plt.subplots_adjust(hspace=.9)
```



[ ]:
```
fig, ax = plt.subplots(2,2, figsize=(12,8))

df["claw_length_cm"].plot.hist(ax=ax[0][0])
ax[0][0].set_title("Claw Length (cm)")

df["size_cm"].plot.hist(ax=ax[0][1])
ax[0][1].set_title("Size (cm)")

df["tail_length_cm"].plot.hist(ax=ax[1][0])
ax[1][0].set_title("Tail Length (cm)")

df["weight_kg"].plot.hist(ax=ax[1][1])
ax[1][1].set_title("Weight (kg)")


plt.subplots_adjust(hspace=0.5)
```


Claw Length (cm)
Frequency
0
200
400
600
800
1000
1200
2
4
6
8
10
12
Size (cm)
Frequency
0
200
400
600
800
1000
1200
1400
50
55
60
65
Tail Length (cm)
Frequency
0
200
400
600
800
1000
−2
0
2
4
6
8
Weight (kg)
Frequency
0
200
400
600
800
1000
1200
2
4
6
8
10