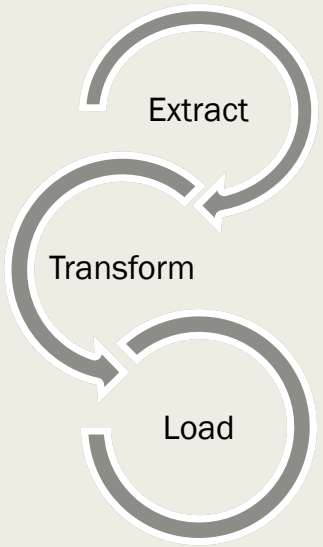
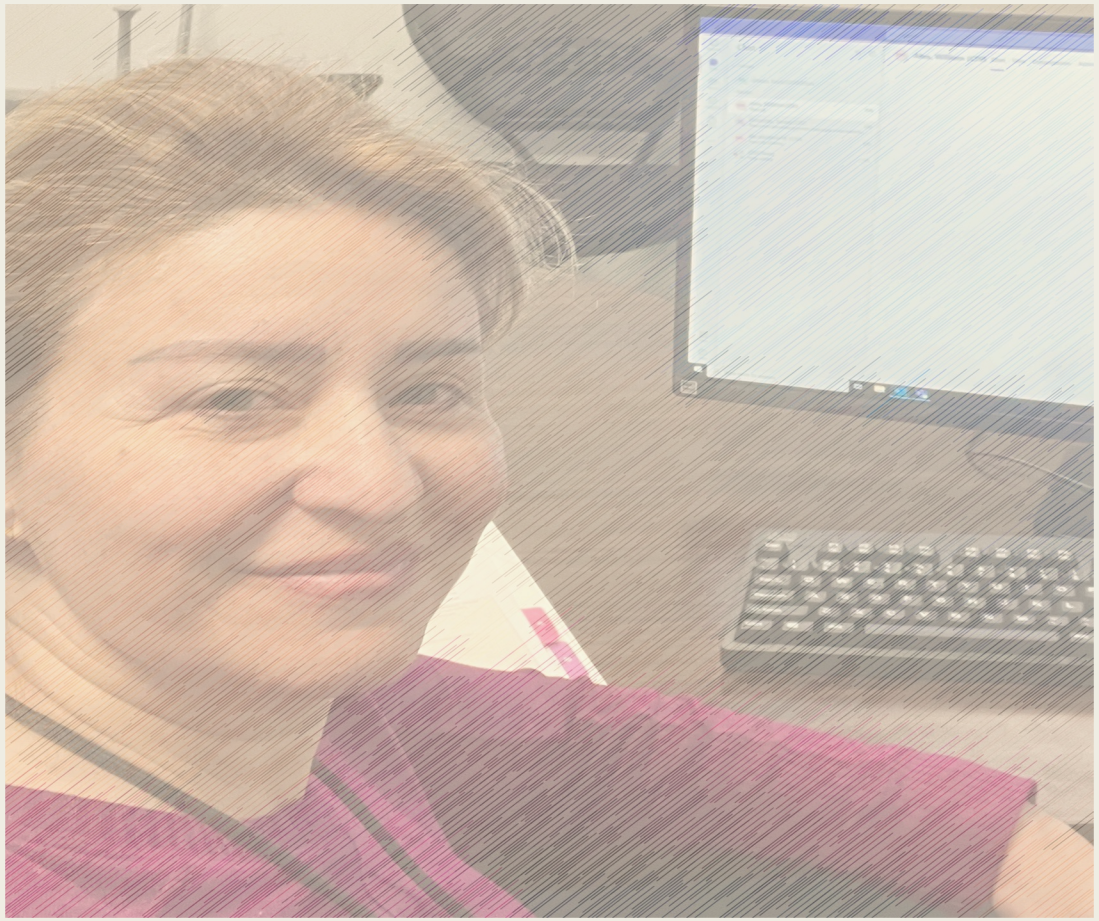


ETL in Nutshell



Rana Ghazzi

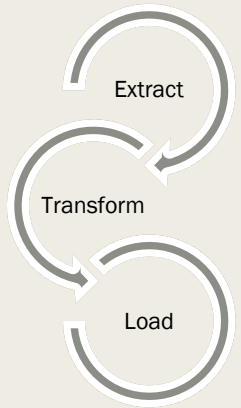
ETL

It’s a data integration process used to combine data from multiple sources into a single, consistent data set for storage in a data warehouse or other target system.

1. **Extract:** This step involves retrieving data from various sources, such as databases, APIs, or flat files. The goal is to gather all relevant data needed for analysis.

2. **Transform:** In this phase, the extracted data is cleaned, formatted, and transformed to meet the requirements of the target system. This can include filtering out errors, converting data types, and applying business rules to ensure consistency and accuracy.

3. **Load:** The final step is loading the transformed data into the target system, such as a data warehouse or data lake. This makes the data available for analysis and reporting.



Why Python:

Python is an excellent choice for ETL processes! It offers several advantages:

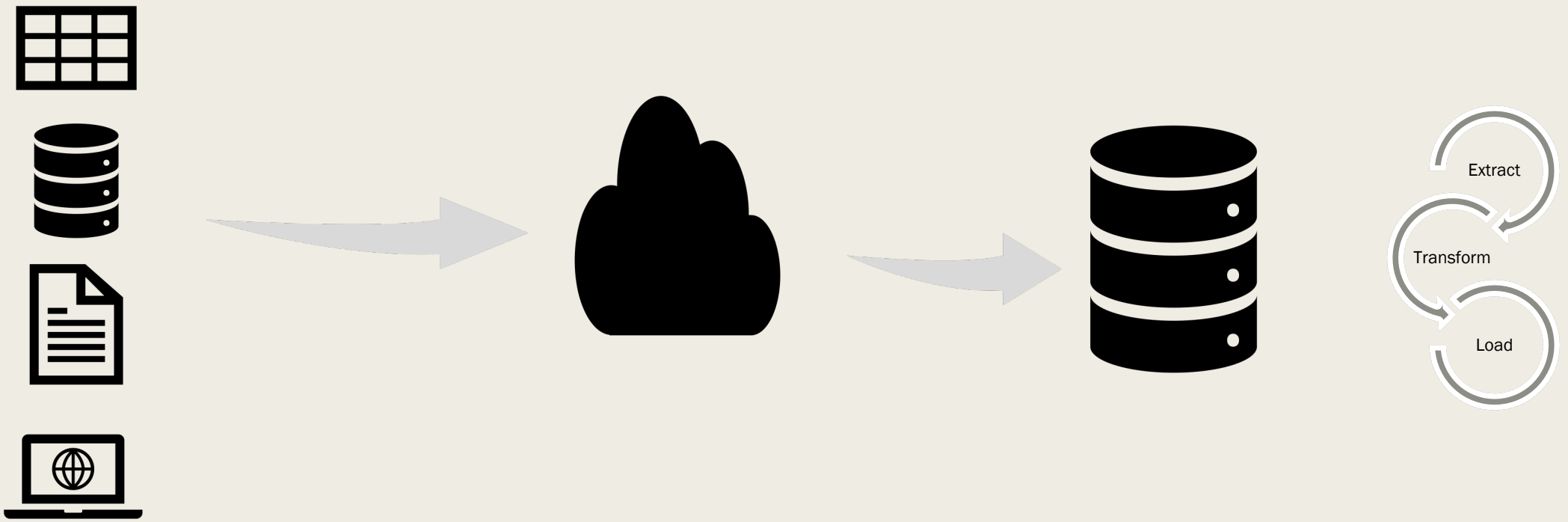
Flexibility: Python is highly flexible and can be used to create custom ETL pipelines tailored to specific needs.

Libraries: There are many powerful libraries available for ETL tasks, such as

Community Support: Python has a large and active community, which means plenty of resources, tutorials, and support are available.

Integration: Python can easily integrate with various data sources and destinations, including databases, APIs, and cloud services.

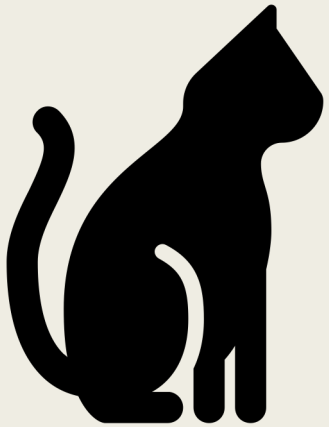
Ease of Learning: Python's syntax is straightforward and easy to learn, making it accessible for both beginners and experienced developers.


Extract
Transform
Load

CASE STUDY:

Our project is for cat lovers, Here’s a brief overview:
The Cat API Project:

Purpose: The Cat API provides access to a vast collection of cat images, breed information, and cat-related facts. It’s designed to help developers easily integrate cat content into their websites or applications.



Features:

- Images: Access to over 60,000 cat images.
- Breeds: Detailed information on various cat breeds.
- Facts: Interesting facts about cats.
- Voting and Favorites: Users can vote on and favorite cat images.

CASE STUDY:

We are utilizing different Python libraries to create a connection to Database / API where we will extract data.

- Explore , clean, and transform our dataset.
- Upload data into a database or save it as a CSV file / OR DATABASE.

Data Source: https://api.thecatapi.com/v1/breeds

Tools To be used: Jupiter Notebook, Python, Pandas, and Postgres Database.

1- Connecting to Data Source:

For this project our data source is API connection that list data about different cats with all info about breeds, origins , names, temperaments, and qualities.

CONNCECTING to DataSource: API AND IMPORTING DATA:

Importing esstentail libraries:

[603]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import requests
from sqlalchemy import create_engine
import psycopg2
import seaborn as sns
from scipy.stats import chi2_contingency
```

[604]:
```
response = requests.get("https://api.thecatapi.com/v1/breeds" ).json()
data=pd.json_normalize(response)
```

Connect to Different Data Sources:

[19]:
```
import psycopg2
import pandas as pd
```

[20]:
```
# Establish a connection
conn = psycopg2.connect(
    database="postgres",
    user="postgres",
    password='[redacted]
    host="localhost",
    [redacted]
)

# Create a cursor object
cur = conn.cursor()

# Execute a query
```

[5]:
```
import requests
import pandas as pd
from sqlalchemy import create_engine
# Project ___2___
response = requests.get("https://api.nobelprize.org/2.1/laureates").json()
df=pd.json_normalize(response)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1 entries, 0 to 0
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   laureates        1 non-null      object
 1   meta.offset      1 non-null      int64 
 2   meta.limit       1 non-null      int64 
 3   meta.count       1 non-null      int64 
 4   meta.terms       1 non-null      object
 5   meta.license     1 non-null      object
```

+ ✂ ⧉ 💾 ▶ ■ ↻ ▶▶ Code

[1]:
```
import pandas as pd
from sqlalchemy import create_engine
import psycopg2
```

[12]:
```
csv_file_path = '/Users/Rana/Desktop/Clean_log.csv'
df = pd.read_csv(csv_file_path)
```

2- EDA: Data Discovery

EDA: a Data exploring

[456]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 67 entries, 0 to 66
Data columns (total 38 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   id                   67 non-null     object
 1   name                 67 non-null     object
 2   cfa_url              43 non-null     object
 3   vetstreet_url        50 non-null     object
 4   vcahospitals_url     42 non-null     object
 5   temperament          67 non-null     object
 6   origin               67 non-null     object
 7   country_codes        67 non-null     object
 8   country_code         67 non-null     object
 9   description          67 non-null     object
```

```
data.columns
```

```
Index(['id', 'name', 'cfa_url', 'vetstreet_url', 'vcahospitals_url',
       'temperament', 'origin', 'country_codes', 'country_code', 'description',
       'life_span', 'indoor', 'lap', 'alt_names', 'adaptability',
       'affection_level', 'child_friendly', 'dog_friendly', 'energy_level',
       'grooming', 'health_issues', 'intelligence', 'shedding_level',
       'social_needs', 'stranger_friendly', 'vocalisation', 'experimental',
       'hairless', 'natural', 'rare', 'rex', 'suppressed_tail', 'short_legs',
       'wikipedia_url', 'hypoallergenic', 'reference_image_id',
       'weight.imperial', 'weight.metric'],
      dtype='object')
```

]:
```
data.shape
```

]: (67, 40)

```
data.duplicated().sum()
```

0

```
data['name'].value_counts().sum()
```

67

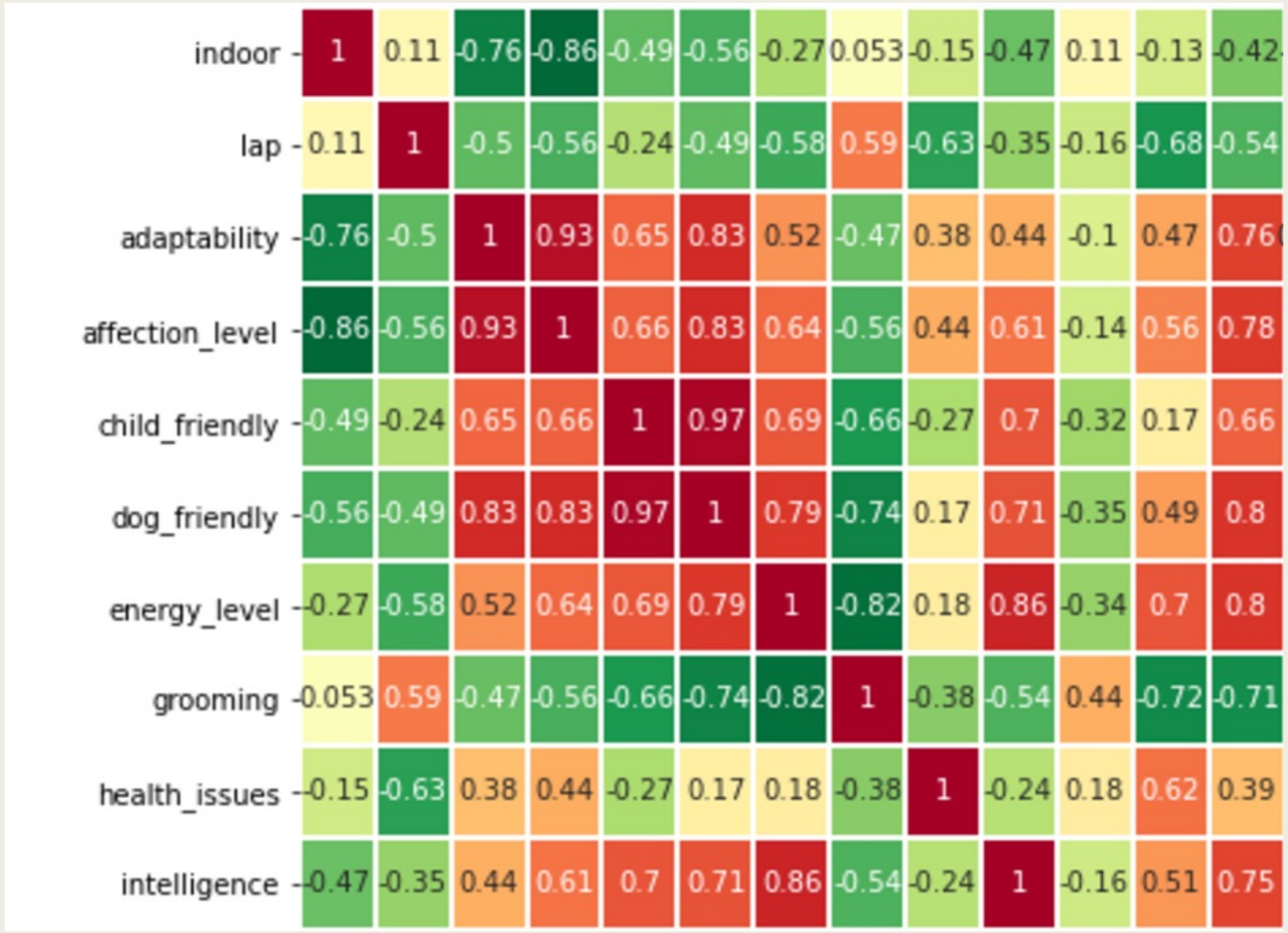
```
data.isna().sum().sort_values(
```

bidability	65
cat_friendly	60
vcahospitals_url	25
cfa_url	24
lap	20
vetstreet_url	17
alt_names	4
reference_image_id	2
wikipedia_url	1

Heatmap AND Correlations

Observation:

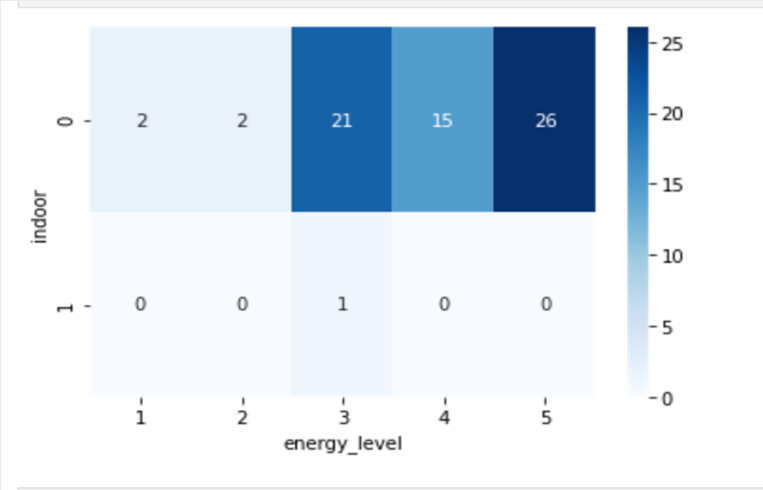
The correlation coefficient values in the heatmap ,that measures the strength and direction of a relationship between two variables, suggests a strong relationships between some variables in this data.

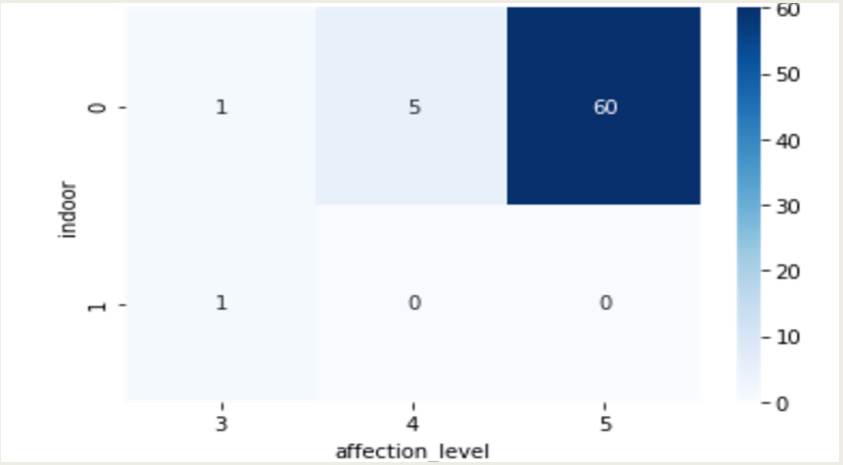


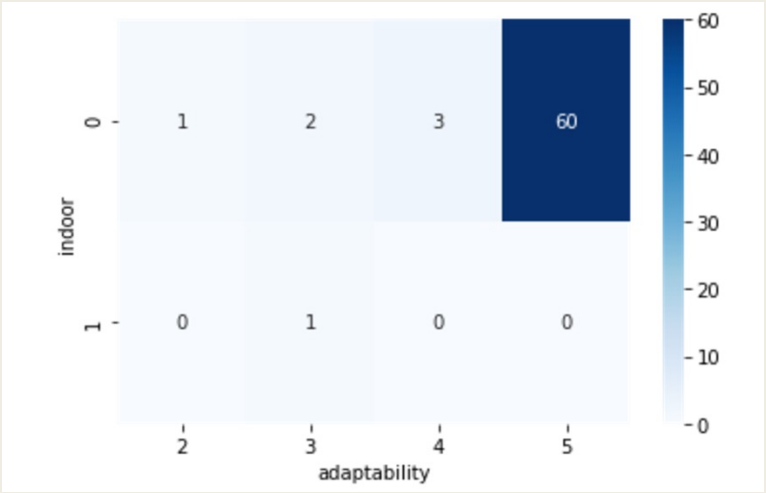
Examples:

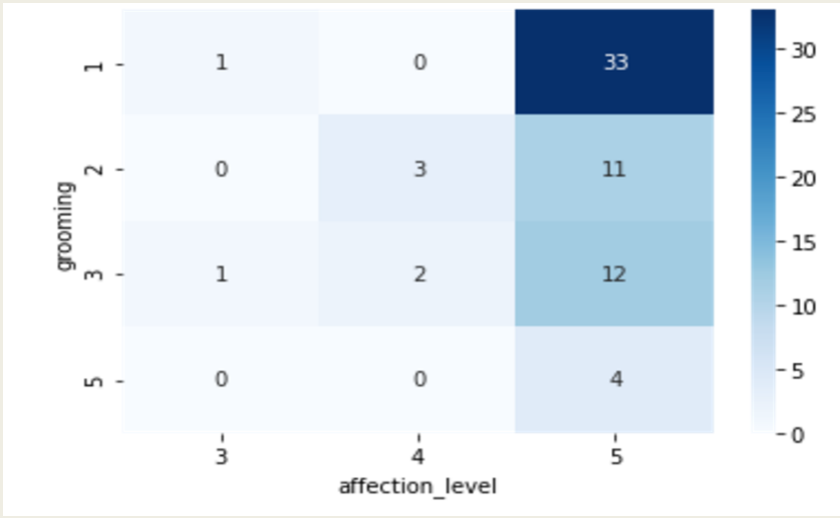
- Indoor & Adaptability,
- Indoor & Affection-Level
- Adaptability & Affection level
- Stanger friendly: Intelligence
- Social-needs: intelligence

As strong associations between some cats qualities is more evident according to the chart shown below:









Insight:

From exploring the columns in this dataset we can conclude the following:

Most of the columns in this dataset are categorical except for few like: weight related measures.

Many of the categorical fields are binominal , ordinal, and few nominal.

The heat map shows strong associations between different variables.

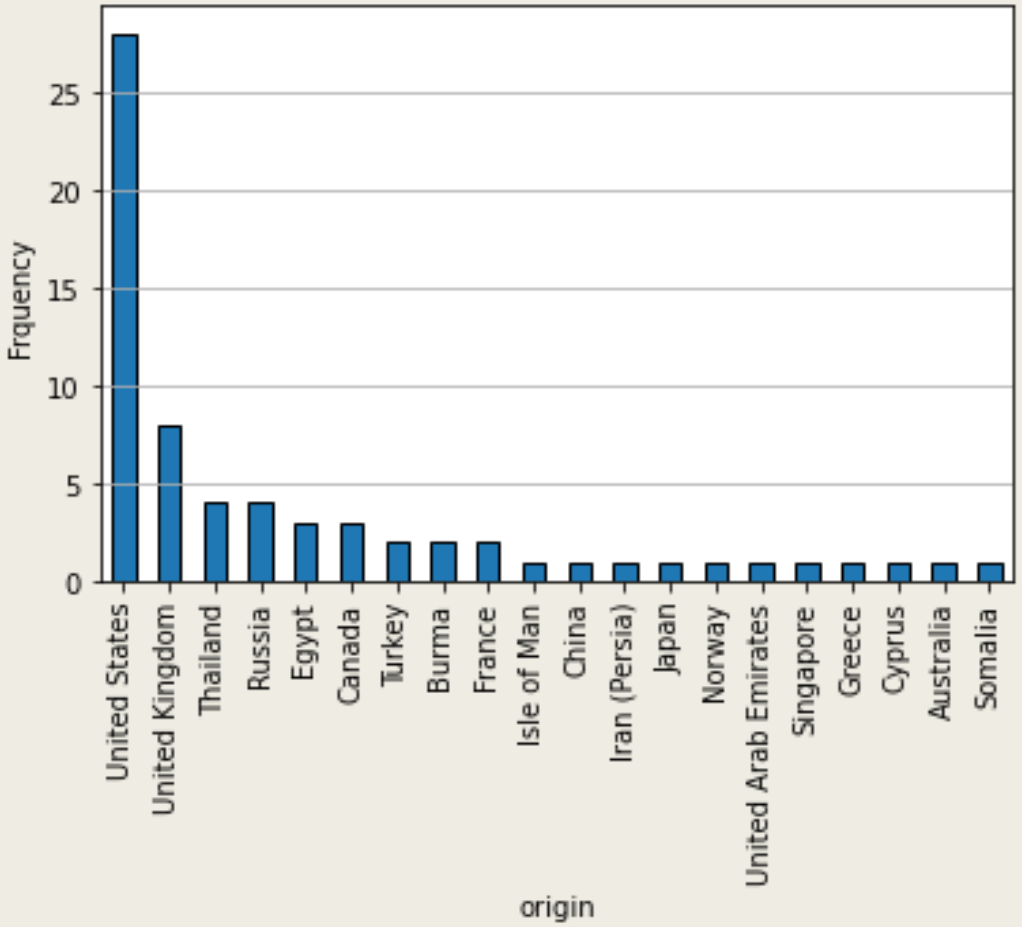
Association between categorical variables will need chi squared test to confirm.(which is out of the scope of this project.)

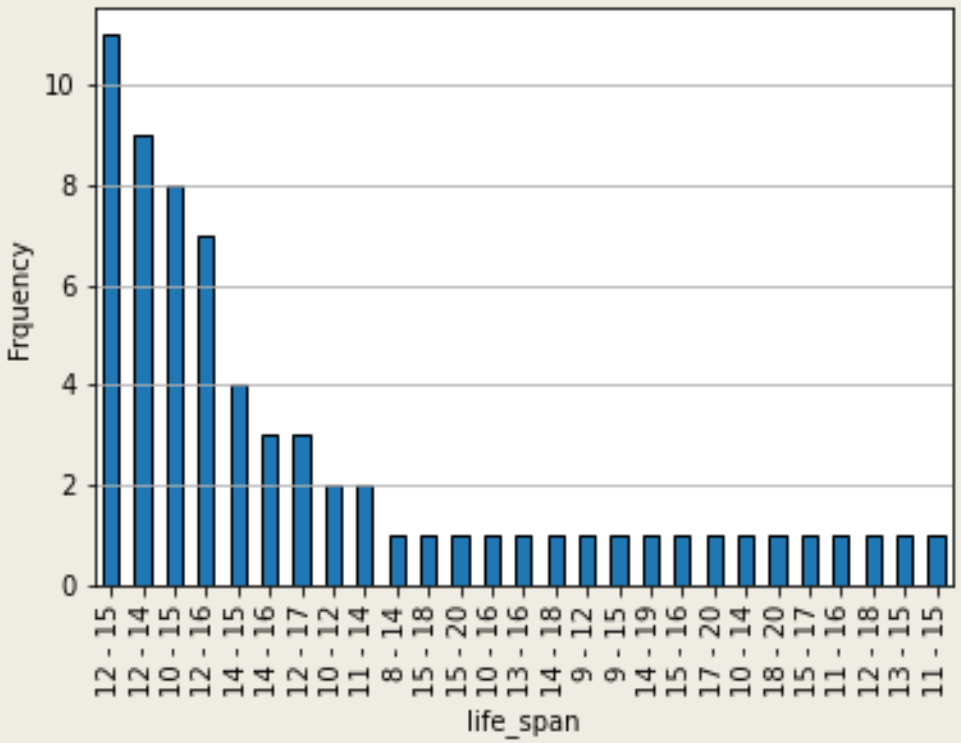
By doing initial exploration to the dataset, we find the following:

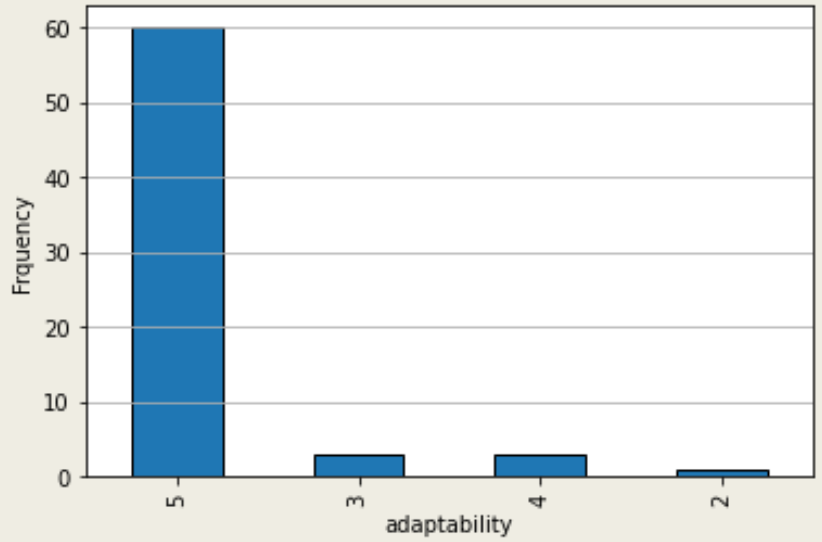
dataset contains 40 columns, 67 records.
The majority of those columns are categorical data type.

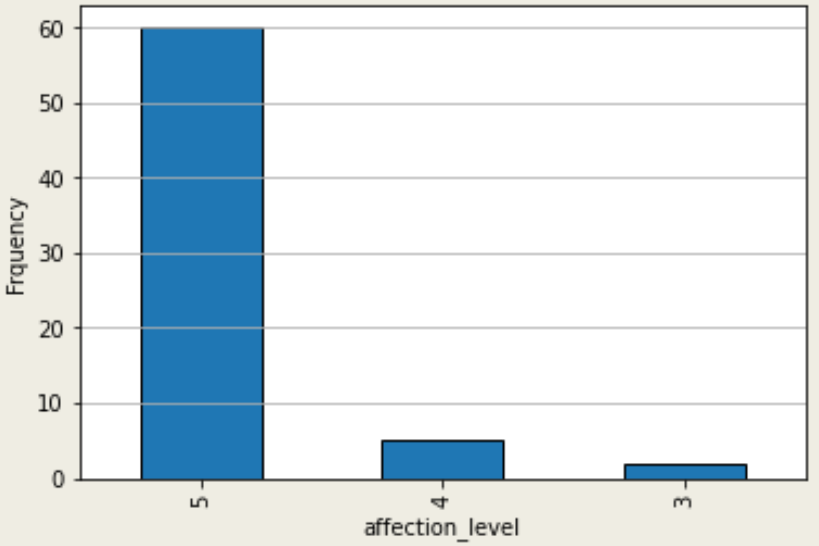
Categorical fields had been encoded in the original source(for analysis reason)

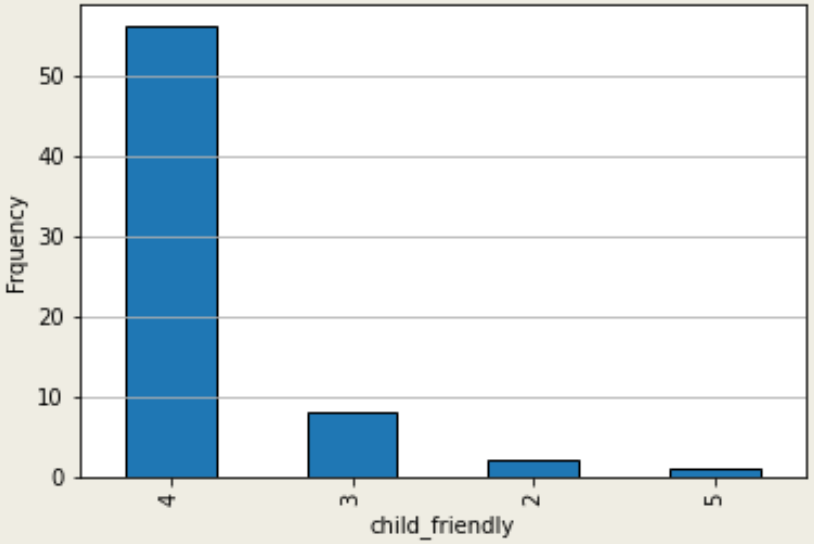
Visual Exploration: Columns' Values & Frequencies:

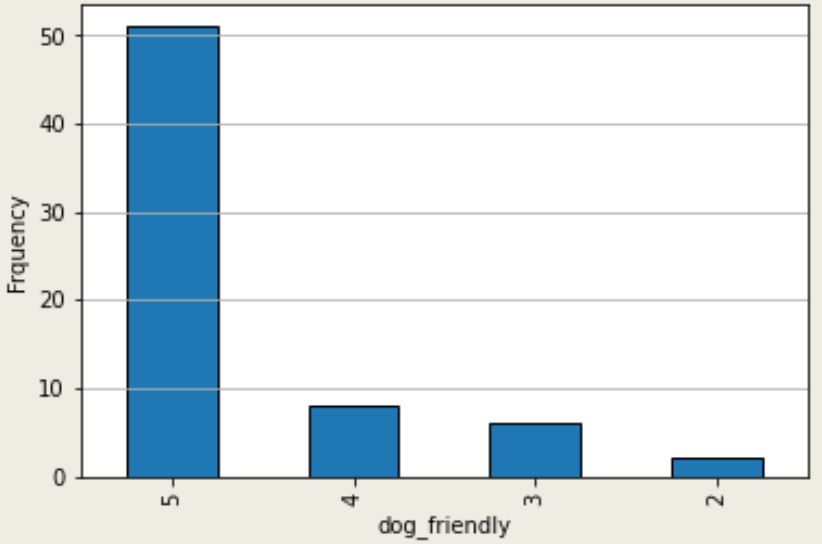


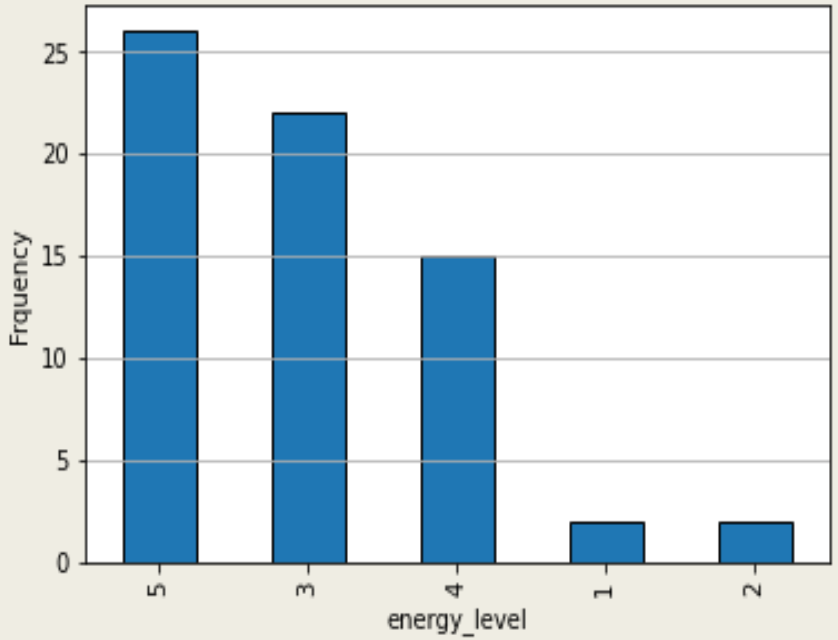


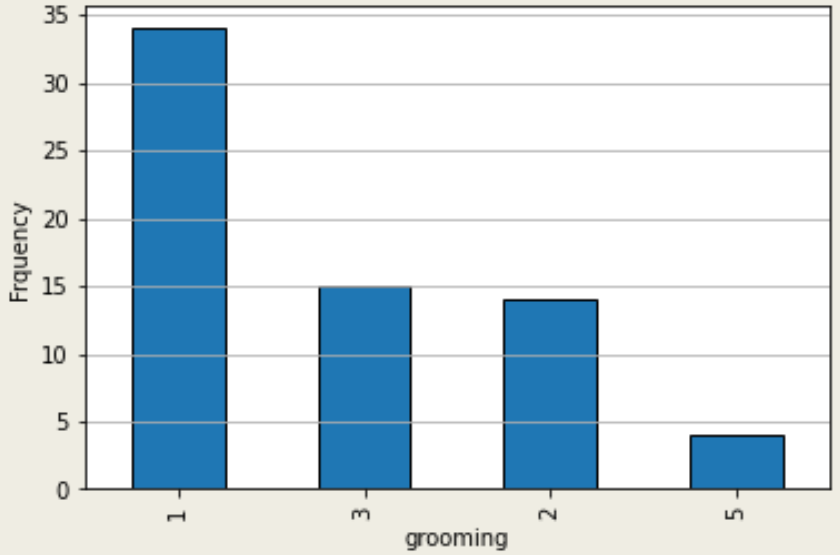

Frquency
adaptability
5
3
4
2

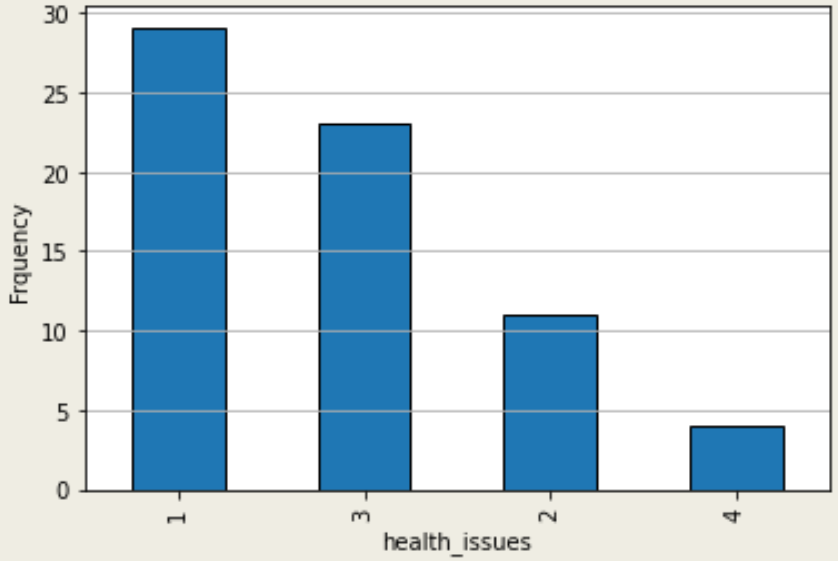

Frquency
affection_level
5
4
3

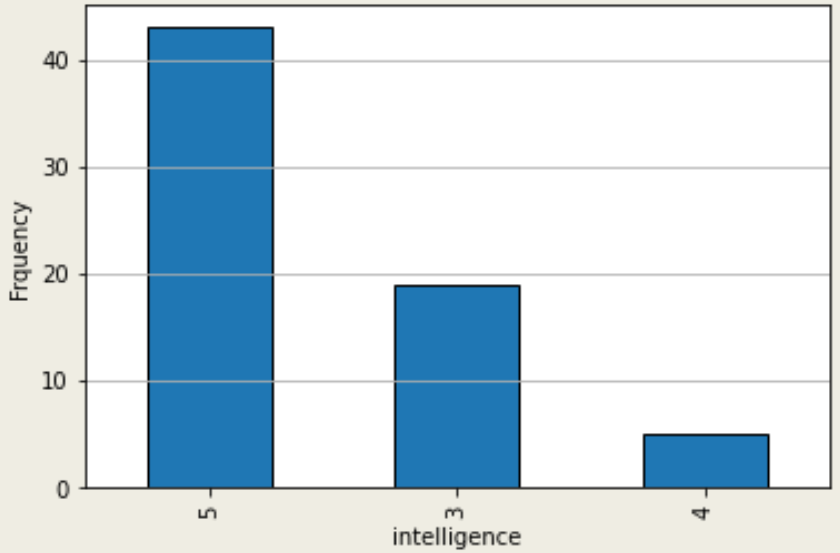

Frquency
child_friendly
4
3
2
5

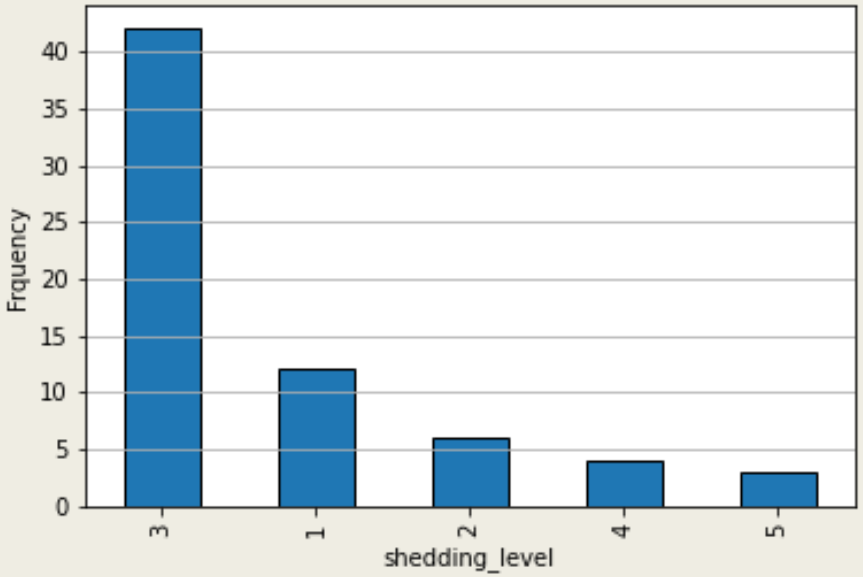

Frquency
dog_friendly
5
4
3
2

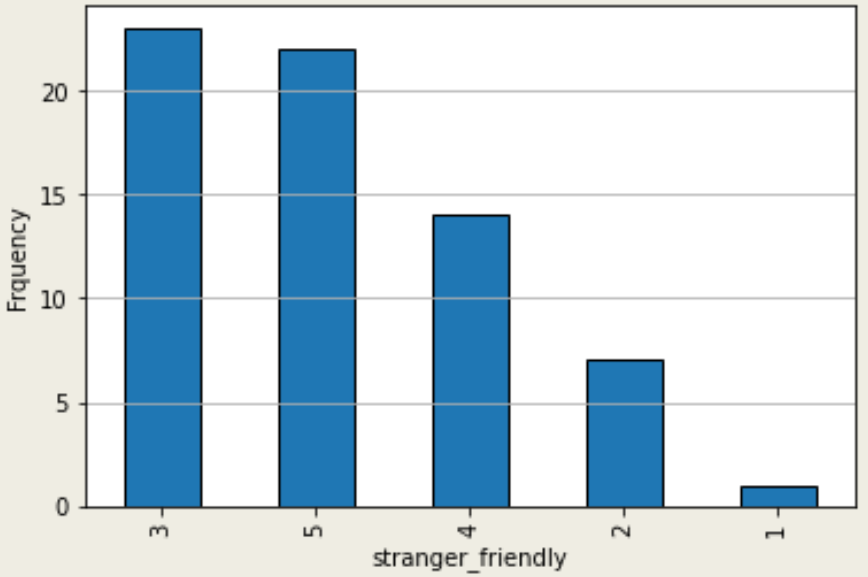

Frquency
25
20
15
10
5
0
5
3
4
1
2
energy_level

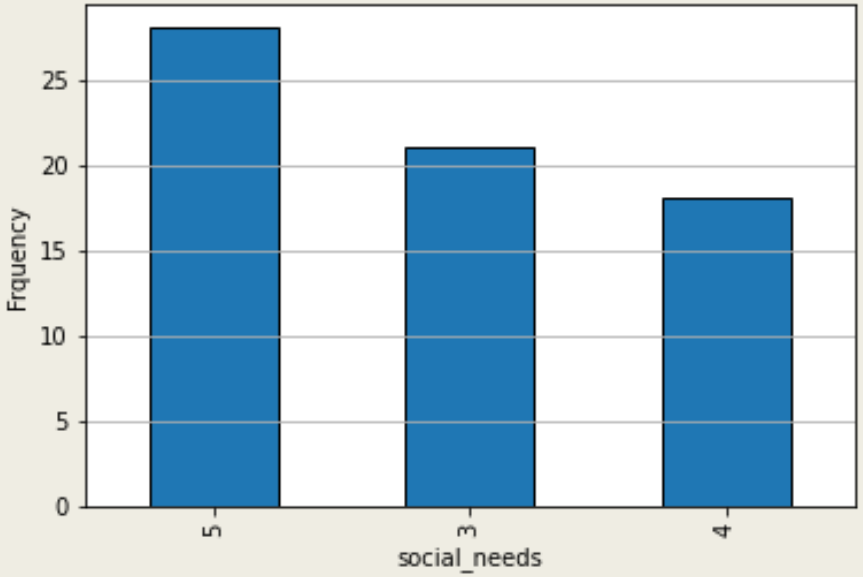

Frquency
35
30
25
20
15
10
5
0
1
3
2
5
grooming

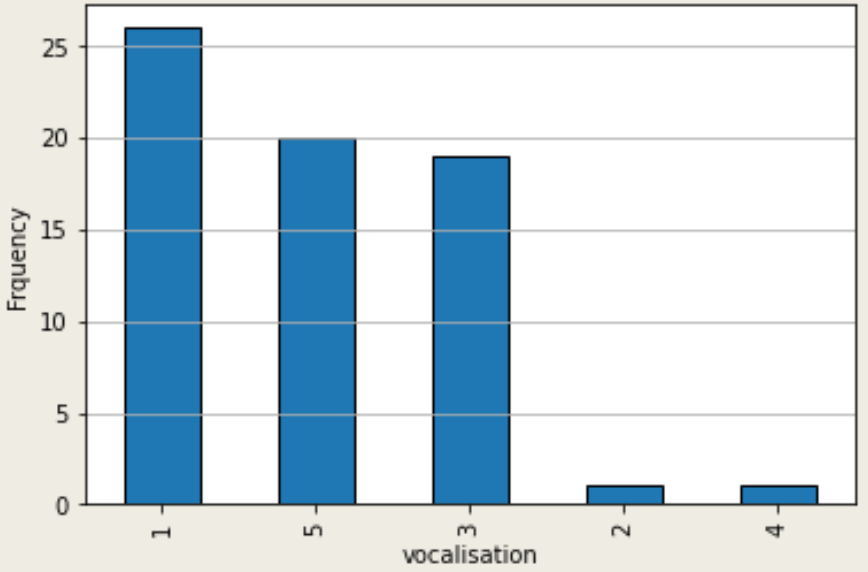

Frquency
30
25
20
15
10
5
0
1
3
2
4
health_issues

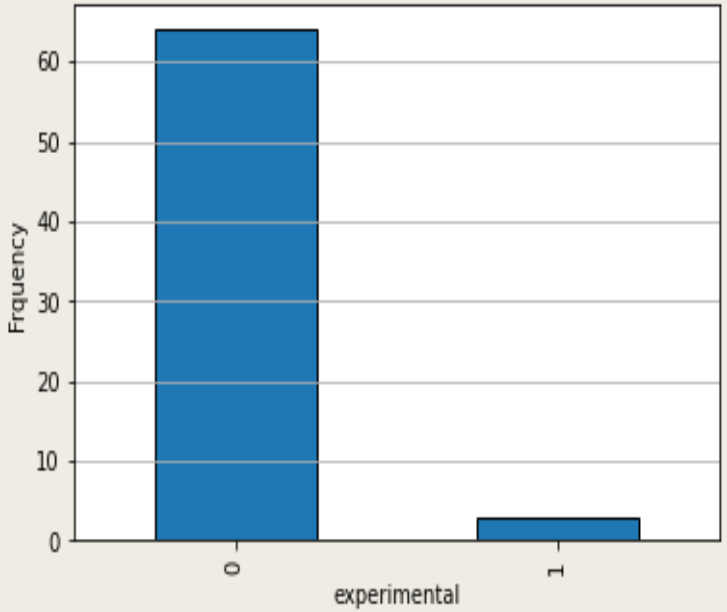

Frquency
40
30
20
10
0
5
3
4
intelligence

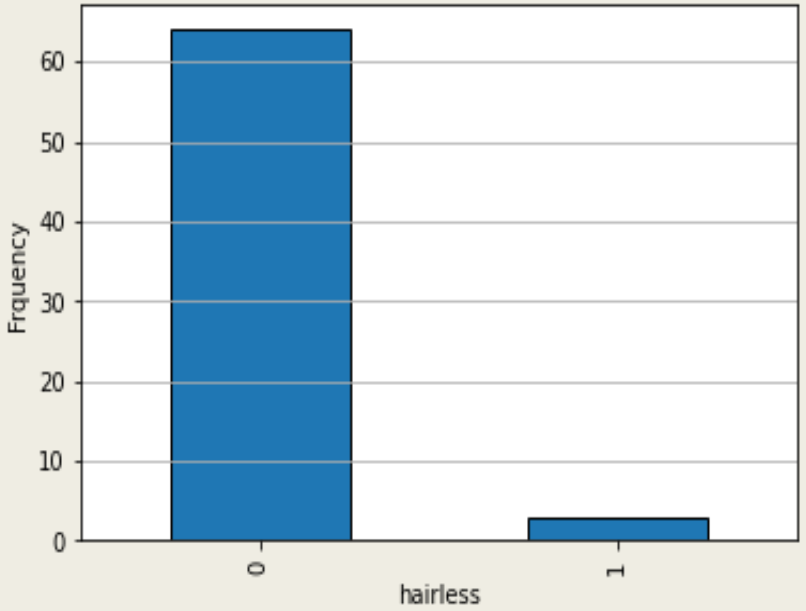

Frquency
0
5
10
15
20
25
30
35
40
3
1
2
4
5
shedding_level

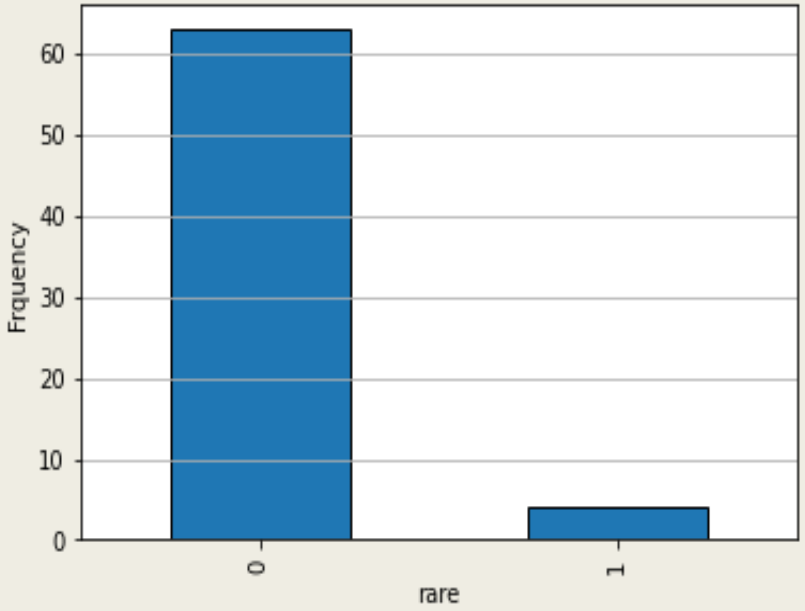

Frquency
0
5
10
15
20
3
5
4
2
1
stranger_friendly

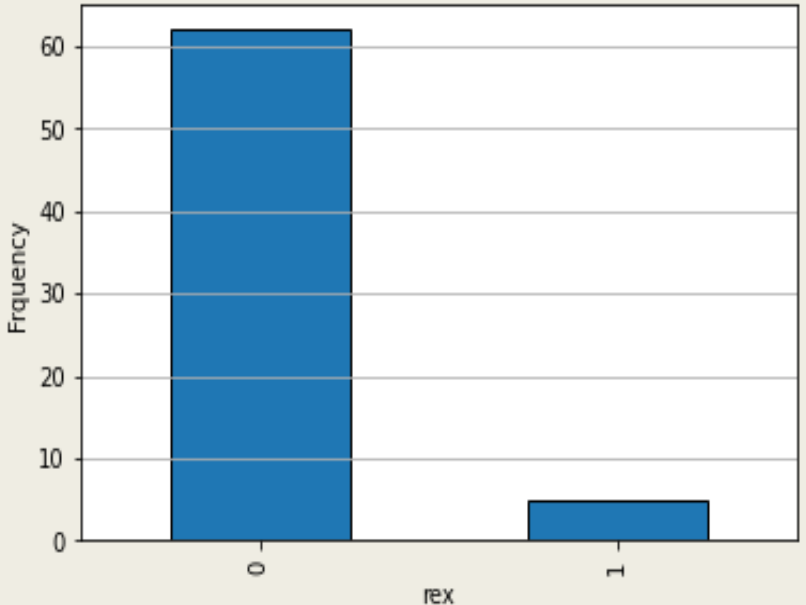

Frquency
0
5
10
15
20
25
5
3
4
social_needs

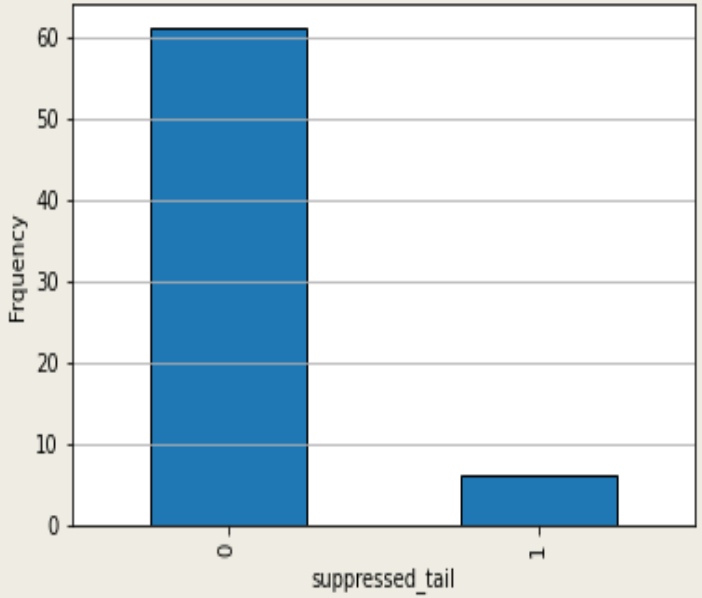

Frquency
0
5
10
15
20
25
1
5
3
2
4
vocalisation

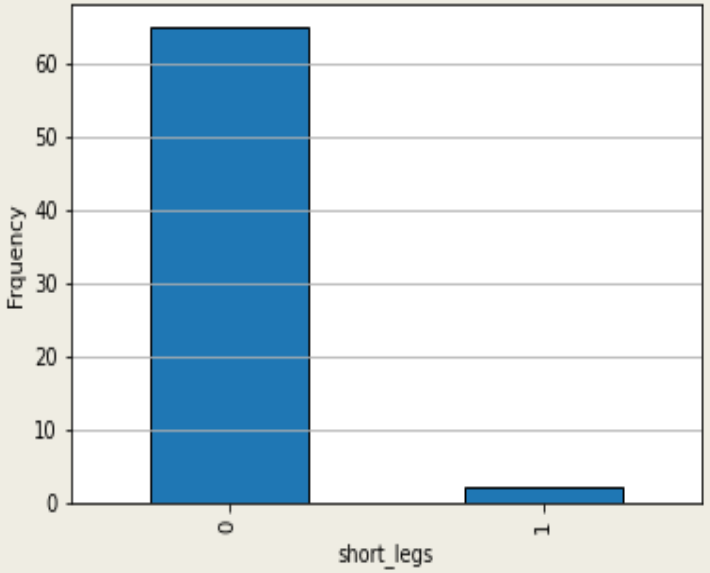

Frquency
60
50
40
30
20
10
0
0
1
experimental

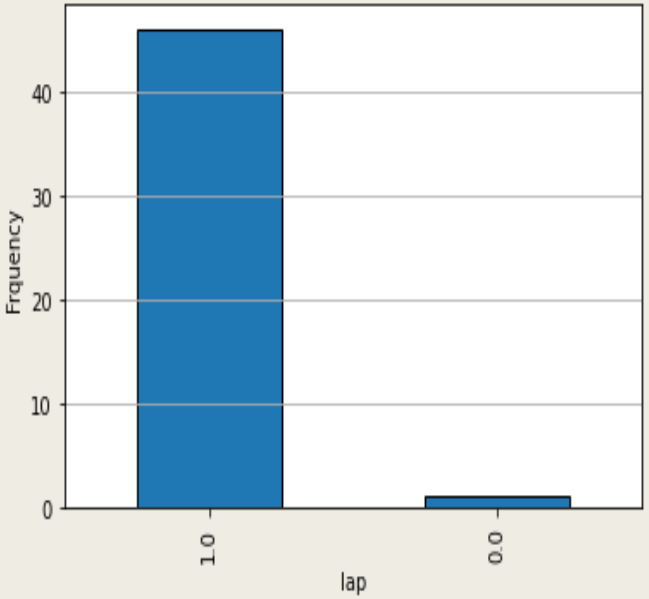

Frquency
60
50
40
30
20
10
0
0
1
hairless

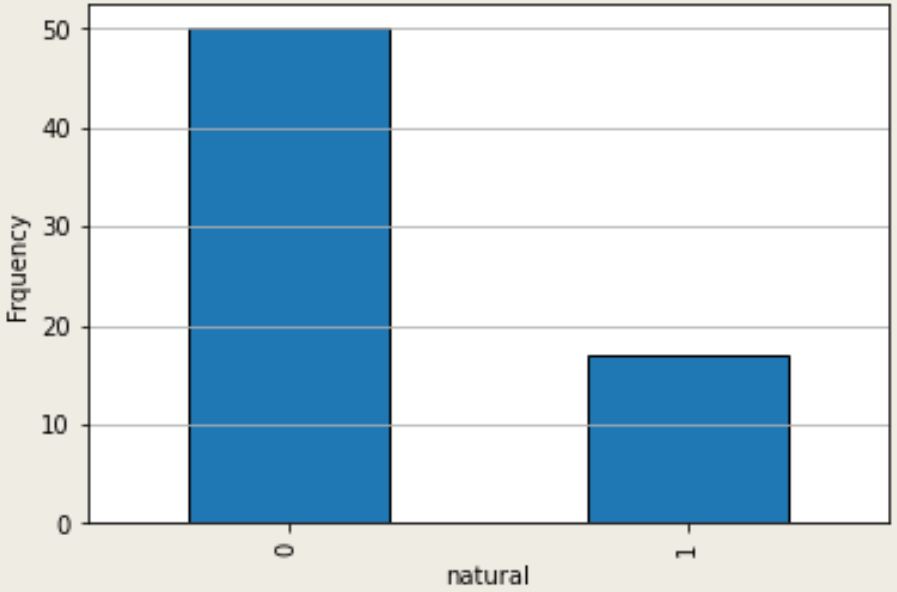

Frquency
60
50
40
30
20
10
0
0
1
rare

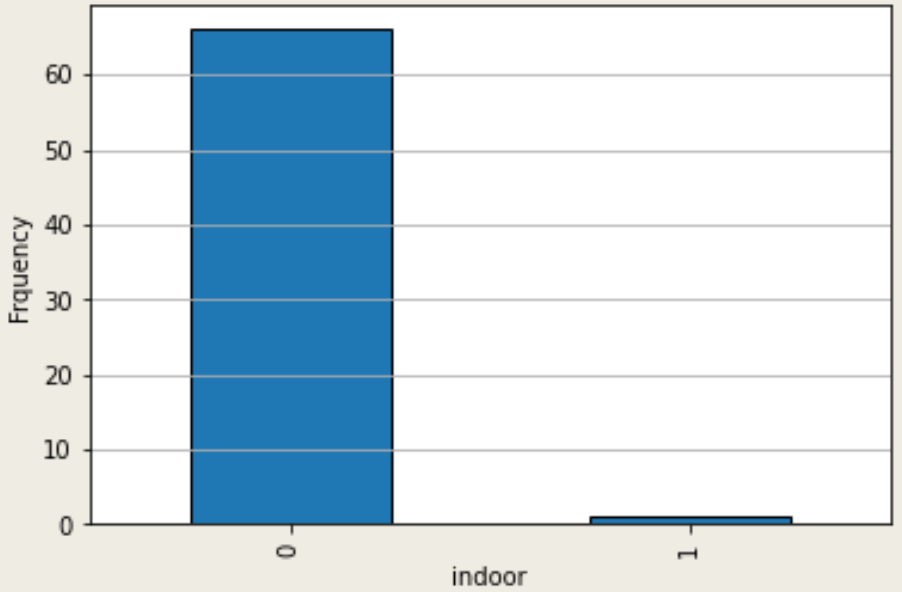

Frquency
60
50
40
30
20
10
0
0
1
rex


Frquency
suppressed_tail
0
1


Frquency
short_legs
0
1


Frquency
lap
1.0
0.0


Frquency
natural
0
1


Frquency
indoor
0
1

3- DATA CLEANING:

Once the data are collected, the next step is to get it ready for analysis. This means cleaning, or scrubbing' it, and is crucial in making sure that you're working with high-quality data.

These include the following steps:

Removing major errors, duplicates, and outliers - all of which are inevitable problems when aggregating data from numerous sources.

Removing unwanted data points-extracting irrelevant observations that have no bearing on your intended analysis.

Data Standardization: fixing typos or layout issues, data types , which will help map and manipulate data more easily.

Filling in major gaps, you might notice that important data are missing. Once you've identified gaps.

Null Values:

We can see that some fields have a null Values of 65, 60 which make those columns useless and can not provide insight.

Replace missing categorical values with the most frequent category (mode) within that variable.

In this Dataset as we can see there are three fields that includes URLs. In my opinion the best approach is just keep is as unknow.

```
data.isna().sum().sort_values(

bidability              65
cat_friendly            60
vcahospitals_url        25
cfa_url                 24
lap                     20
vetstreet_url           17
alt_names                4
```

```
]: data.shape

]: (67, 40)
```

```
[717]: data['vetstreet_url']

[717]: 0            http://www.vetstreet.com/cats/abyssinian
       1            http://www.vetstreet.com/cats/aegean-cat
       2      http://www.vetstreet.com/cats/american-bobtail
       3         http://www.vetstreet.com/cats/american-curl
       4    http://www.vetstreet.com/cats/american-shorthair
       5     http://www.vetstreet.com/cats/american-wirehair
       6
```

Duplicates vales:

We have no duplicated records in our dataset.

```
data.duplicated().sum()
```

```
0
```

Data Type:

By doing initial exploration to the dataset, we find the following:

Dataset contains 40 columns, 67 records.

The majority of those columns are categorical data type.

Categorical fields had been encoded in their original source (for analysis reason)

20	health_issues	67 non-null	int64
21	intelligence	67 non-null	int64
22	shedding_level	67 non-null	int64
23	social_needs	67 non-null	int64
24	stranger_friendly	67 non-null	int64
25	vocalisation	67 non-null	int64
26	experimental	67 non-null	int64
27	hairless	67 non-null	int64
28	natural	67 non-null	int64
29	rare	67 non-null	int64
30	rex	67 non-null	int64
31	suppressed_tail	67 non-null	int64
32	short_legs	67 non-null	int64
33	wikipedia_url	67 non-null	object
34	hypoallergenic	67 non-null	int64
35	reference_image_id	67 non-null	object
36	weight.imperial	67 non-null	object
37	weight.metric	67 non-null	object
38	cat_friendly	67 non-null	object
39	bidability	67 non-null	object

For more about dealing with categorical data you can read the following two slides.

Dealing with Categorical Data (1):

Identify categorical features:

Identifying which columns in your dataset contain categorical data, like text descriptions or labels .

Handle missing values:
Replace missing categorical values with the most frequent category (mode) within that variable.

Encoding methods:
Label encoding: Assign a unique integer to each category, suitable when there's an inherent order between categories (like size: small, medium, large).

One-hot encoding: Create new binary features for each category, where only one feature is ""for a given category, useful when order is not important.

Consider cardinality:
Low cardinality: If there are few unique categories, one-hot encoding is often suitable.

High cardinality: For many unique categories, consider techniques like target encoding or frequency encoding to reduce dimensionality.

Dealing with Categorical Data (2):

A chi-square test is used to assess whether there is a significant association between two categorical variables, while a Pearson correlation measures the strength of a linear relationship between two continuous variables; essentially.

A chi-square test looks for relationships between categories, while a Pearson correlation examines how two numerical variables change together in a linear fashion

There are three metrics that are commonly used to calculate the correlation between categorical variables:

1. Tetrachoric Correlation: Used to calculate the correlation between binary categorical variables.

2. Polychromic Correlation: Used to calculate the correlation between ordinal categorical variables.

3. Cramer's V: Used to calculate the correlation between nominal categorical variables.
Note: Nominal data is a type of data that categorizes variables into distinct groups without any inherent order or ranking

Outliers:

There are no outlier detection methods for categorical data.

For an outlier to exist there must be a measure of distance between the items.
Removing outliers involves excluding data points significantly deviating from the norm

Removing outliers influences the mean, reducing its sensitivity to extreme values and providing a more representative measure of central tendency.

Common techniques include visualization tools (box plots, scatter plots), mathematical methods (Z-scores, IQR), and threshold-based filtering.

3- DATA LOADING:

Connecting to the Target Database (Postgres)

```
db_username = "postgres"
db_password ="[redacted]
db_host = "localhost"
db_port = [redacted]
db_name = [redacted]                    .
engine = create_engine(f'postgresql://{db_use
table_name = 'cats'
data.to_sql(table_name, engine, if_exists='ap

print("Data inserted successfully!")
```

```
Data inserted successfully!
```

```
1  SELECT * FROM public.cat
2
```

Data Output | Explain | Messages | Notifications

	id text	name text	cfa_url text	vetstreet_url text
1	abys	Abyssinian	http://cfa.org/Breeds/BreedsAB/Abyssinian.aspx	http://www.vetstreet.com/cats/abyssinian
2	aege	Aegean	[null]	http://www.vetstreet.com/cats/aegean-cat
3	abob	American Bobtail	http://cfa.org/Breeds/BreedsAB/AmericanBobtail.aspx	http://www.vetstreet.com/cats/american-bobtail
4	acur	American Curl	http://cfa.org/Breeds/BreedsAB/AmericanCurl.aspx	http://www.vetstreet.com/cats/american-curl
5	asho	American Shorthair	http://cfa.org/Breeds/BreedsAB/AmericanShorthair.aspx	http:// http://www.vetstreet.com/cats/american-cu
6	awir	American Wirehair	http://cfa.org/Breeds/BreedsAB/AmericanWirehair.aspx	http://www.vetstreet.com/cats/american-wirehair
7	amau	Arabian Mau	[null]	[null]
8	amis	Australian Mist	[null]	[null]
9	bali	Balinese	http://cfa.org/Breeds/BreedsAB/Balinese.aspx	http://www.vetstreet.com/cats/balinese
10	bamb	Bambino	[null]	[null]
11	beng	Bengal	http://cfa.org/Breeds/BreedsAB/Bengal.aspx	http://www.vetstreet.com/cats/bengal
12	birm	Birman	http://cfa.org/Breeds/BreedsAB/Birman.aspx	http://www.vetstreet.com/cats/birman
13	bomb	Bombay	http://cfa.org/Breeds/BreedsAB/Bombay.aspx	http://www.vetstreet.com/cats/bombay
14	bslo	British Longhair	[null]	[null]
15	bsho	British Shorthair	http://cfa.org/Breeds/BreedsAB/BritishShorthair.aspx	http://www.vetstreet.com/cats/british-shorthair

Thank You.

https://ranaghazzi.com