analysis

July 4, 2024

1 Introduction

Chile has a flora rich in biodiversity given its varied geographical conditions, e.i. a territory with at least four types of climate, surrounded by a mountain range and the Pacific Ocean.

Many of these species continue to be studied, which has led to projects to collect functional traits of Chilean plants, such as Rasgos-CL and HerbarioDigital.

In this project, my objective is to analyze the biases in the data sets mentioned above. In other words, I seek to analyze and explain as far as possible the lack of data in some of the registered species, hoping to find probable causes in the level of vulnerability of the species, their geographical distribution in the national territory, or their type, i.e. native or endemic.

To achieve this, I will use the datasets provided by both repositories: I will download CSV files from the Rasgos-CL GitHub repository and obtain the data from the HerbarioDigital through its public API.

2 Data Relevance

Why Rasgos-CL? Rasgos-CL is an excellent repository of public data on Chilean flora species. Specifically, it provides flora's functional trait data such as type of fruit and height. One point to consider is that we will not use your data per se, but rather extract metadata from the repository, e.i. we are not interested in the traits collected, but rather the number of traits documented.

Why HerbarioDigital? This repository, although not as rich in data as Rasgos-CL, does provide us with data on the species studied such as their conservation status or the regions they inhabit.

Other possible data repositories one might use to complement information are the TRY Plant Trait Database and the GIFT database. Those are excellent sources of plant trait data, but unfortunately the first one is too cumbersome to use and one must be registered as a user, while the latter only provides access through an R package.

3 Background

Chile is home to a wide variety of endangered species and the best way to protect them is to understand them better. To help strengthen these efforts, it is essential to know why we know little about some of these species, whether due to difficulty in accessing their habitat or due to their small population.

With this objective in mind, I am preparing to unify the data sets to be able to observe: 1. The number of traits recorded per species. 2. The location distribution of the species. 3. The conservation status of the species. 4. The type of species, e.i. native or endemic. 5. Minimum and maximum height if the species.

With the data described above, I hope to be able to analyze the correlation between the number of traits recorded and the conservation characteristics of the species to better understand the causes of lack of knowledge or biases about them.

4 Import needed libraries

Different tasks need to be done to successfully analyze the collected data. For data manipulation and conversion, we'll use `pandas`, `numpy`, and `scikit-learn`. While for plotting, `matplotlib` will suffice.

As we need to get the data from different sources, we will use `pandas` to collect Rasgos-CL repositories, while I've prepared a download and processing pipeline for Herbario Digital's data.

You can check the herbario's data pipeline at `herbario.py`.

[ ]:
```python
# Data manipulation libraries
from sklearn.preprocessing import MaxAbsScaler, OrdinalEncoder
import pandas as pd
import numpy as np

# Plotting libraries
import matplotlib.pyplot as plt

# Typings for better readability
from typing import List, Dict

# Custom data retrieval pipeline
from herbario import pipeline
```

4.1 Set basic configuration

Some basic configuration migh be needed, according to some of the libraries documentation.

[ ]:
```python
# Recommended on documentation: https://pandas.pydata.org/pandas-docs/stable/
↪user_guide/indexing.html#returning-a-view-versus-a-copy
pd.options.mode.copy_on_write = True
```

5 Utility functions

As some tasks need to be repeated with different data, we can follow a DRY approach by creating some utility functions, specially for plotting.

[ ]:
```python
def plot_bar(data: pd.DataFrame=None, x=None, y=None, title="", xlabel="",␣
↪ylabel="", legend: List[str]=None, color: List[str]=None, xticks=None):
    """
    Plots bar chart from df data.

    The DataFrame used should have just one column and a value-defined index␣
↪or, if None, x and y values should be ArrayLike.
    """
    # Create main plot or raise an error if data is not in right format
    if data is not None:
        data.plot.bar(color=color)
    elif x is not None and y is not None:
        plt.bar(x=x, height=y, color=color)
    else:
        raise ValueError("must provide data or x and y args")

    # Set extra properties
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    if legend is not None:
        plt.legend(legend)
    plt.xticks(rotation=90)
    if xticks is not None:
        plt.xticks(xticks)

    # Display the plot
    plt.show()
```

[ ]:
```python
def plot_scatter(x,y, title="", xlabel="", ylabel=""):
    """
    Plots histogram chart from dataframe data.
    """
    # Create main plot
    plt.scatter(x=x, y=y)

    # Set extra properties
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)

    # Display the plot
    plt.show()
```

6 Getting the data

Now, let ourselves get both Rasgos-CL and Herbario Digital's data repositories ready for analysis.

6.1 Get Rasgos-CL data

The datasets are located at GitHub.com, inside a public repository.

[ ]:
```
traits_url = "https://raw.githubusercontent.com/dylancraven/Rasgos-CL/main/Data/
    ↪RasgosCL_aggregatedspp.csv"
try:
    traits_df = pd.read_csv(traits_url)
except Exception as err:
    print(f"Error when downloading: {err}")
```

We also need a secondary dataset for geographical references:

[ ]:
```
geo_url = "https://raw.githubusercontent.com/dylancraven/Rasgos-CL/main/Extra/
    ↪Chile_spp_distrib.csv"
try:
    geo_df = pd.read_csv(geo_url)
except Exception as err:
    print(f"Error when downloading: {err}")
```

6.2 Get HerbarioDigital's data

As stated before, we'll use our custom data retrieval pipeline to get a `pandas.DataFrame`.

[ ]:
```
herbario_species = pipeline()
```

```
Reusing last species file
```

7 Exploration

We'll explore the data first by understanding the shape and general information of every dataset.

7.1 Rasgos-CL species data exploration

First of all, we'll check Rasgos-CL species data.

Some general information on the shape and statistics:

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8643 entries, 0 to 8642
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   accepted_species  8643 non-null   object
```

```
 1   traitValue        8643 non-null   object
 2   obs               8413 non-null   float64
 3   traitName         8643 non-null   object
 4   agreement         7256 non-null   float64
 5   traitUnit         8643 non-null   object
dtypes: float64(2), object(4)
memory usage: 405.3+ KB
```

The dataset contains `empty` data, e.i. `NaN` values, in the columns `obs` and `agreement`.

[ ]:
```
traits_df.describe()
```

[ ]:
```
               obs    agreement
count  8413.000000  7256.000000
mean      3.020088     0.986776
std       3.956636     0.057783
min       1.000000     0.666667
25%       1.000000     1.000000
50%       2.000000     1.000000
75%       4.000000     1.000000
max     116.000000     1.000000
```

[ ]:
```
# Checking a randomized sample
traits_df.sample(5)
```

[ ]:
```
             accepted_species    traitValue  obs             traitName  \
5960  Myrceugenia lanceolata         Shrub  1.0           Growth_form
3827   Geoffroea decorticans  Endozoochory  1.0  Dispersal_syndrome_3
7646   Senecio chanaralensis     Evergreen  1.0            Leaf_habit
2133      Chersodoma candida  Dry_pericarp  3.0           Fruit_type_2
123         Adesmia balsamica  Simple_fruit  4.0           Fruit_type_1

      agreement                                          traitUnit
5960        1.0                      Tree, Shrub_SmallTree, Shrub
3827        1.0                Ballistic, Endozoochory, Epizoochory
7646        1.0                      Deciduous, Evergreen, Variable
2133        1.0                                Dry_pericarp, Fleshy
123         1.0  Compound_fruit, None, Pseudo_fruit, Simple_fru…
```

7.2 Rasgos-CL geographical data exploration

Now, let's check the geographical data, which is more like a distribution dictionary:

[ ]:
```
# Gain insights by looking at the first 5 values
geo_df.head()
```

[ ]:
```
   accepted_species region  presencia
0  Acrisione cymosa    AIS          1
```

```
1  Acrisione cymosa    ANT          0
2  Acrisione cymosa    ARA          1
3  Acrisione cymosa    ATA          0
4  Acrisione cymosa    AYP          0
```

This dataset is used to check the presence of every specie in different locations of Chile, where `1` means present, and `0` absent.

Also, the `region` column is encoded, so we probably will need to translate it into something useful.

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12812 entries, 0 to 12811
Data columns (total 3 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   accepted_species  12812 non-null  object
 1   region            12812 non-null  object
 2   presencia         12812 non-null  int64
dtypes: int64(1), object(2)
memory usage: 300.4+ KB
```

Data looks fine, as every column has valid data.

[ ]:
```
geo_df.region.unique()
```

[ ]:
```
array(['AIS', 'ANT', 'ARA', 'ATA', 'AYP', 'BIO', 'COQ', 'IPA', 'JFE',
       'LBO', 'LLA', 'LRI', 'MAG', 'MAU', 'NUB', 'RME', 'TAR', 'VAL'],
      dtype=object)
```

What's interesting here is that there are 16 possible regions instead of 15 (Chile only has 15 regions).

This dataset contemplates some of Chile's most important islands, i.e. Easter and Juan Fernández islands, as independent regions.

7.3 Herbario Digital data exploration

Very similar to Rasgos-CL species dataset, this dataset contains information about every specie characteristics.

[ ]:
```
# Gain insights by looking at the first 5 values
herbario_species.head()
```

[ ]:
```
   Unnamed: 0    id                      scientific_name              habit  \
0           0  1583                     Acrisione cymosa  Bush or Small tree
1           1  5567                Acrisione denticulata               Bush
2           2  1585  Acrisione denticulata var. pilota               Bush
3           3  3523                 Adenopeltis serrata               Bush
4           4  3587                      Adesmia aphylla               Bush
```

```
    status  conservation_state  maximum_height  minimum_height  Araucanía  \
0  Endemic  Not Evaluated (NE)          1100.0             0.0          1
1   Native  Not Evaluated (NE)             NaN             NaN          0
2  Endemic  Not Evaluated (NE)          1200.0             0.0          1
3  Endemic  Not Evaluated (NE)           900.0             5.0          0
4  Endemic  Not Evaluated (NE)          3000.0          1800.0          0

   Maule  …  Libertador Bernardo O'Higgins  Arica y Parinacota  Los Ríos  \
0      0  …                              0                   0         1
1      0  …                              0                   0         0
2      1  …                              1                   0         0
3      1  …                              1                   0         0
4      0  …                              0                   0         0

   Ñuble  Coquimbo  Los Lagos  Magallanes  Bío-Bío  Valparaíso  Aysén
0      0         0          1           0        1           1      1
1      0         0          0           0        0           0      0
2      1         1          1           0        1           1      0
3      1         1          0           0        1           1      0
4      0         1          0           0        0           0      0

[5 rows x 25 columns]
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 742 entries, 0 to 741
Data columns (total 25 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Unnamed: 0                     742 non-null    int64
 1   id                             742 non-null    int64
 2   scientific_name                742 non-null    object
 3   habit                          742 non-null    object
 4   status                         737 non-null    object
 5   conservation_state             742 non-null    object
 6   maximum_height                 690 non-null    float64
 7   minimum_height                 720 non-null    float64
 8   Araucanía                      742 non-null    int64
 9   Maule                          742 non-null    int64
 10  Atacama                        742 non-null    int64
 11  Antofagasta                    742 non-null    int64
 12  Juan Fernández                 742 non-null    int64
 13  Tarapacá                       742 non-null    int64
 14  Metropolitana                  742 non-null    int64
 15  Libertador Bernardo O'Higgins  742 non-null    int64
```

```
 16   Arica y Parinacota             742 non-null    int64
 17   Los Ríos                       742 non-null    int64
 18   Ñuble                          742 non-null    int64
 19   Coquimbo                       742 non-null    int64
 20   Los Lagos                      742 non-null    int64
 21   Magallanes                     742 non-null    int64
 22   Bío-Bío                        742 non-null    int64
 23   Valparaíso                     742 non-null    int64
 24   Aysén                          742 non-null    int64
dtypes: float64(2), int64(19), object(4)
memory usage: 145.0+ KB
```

`NaN` values at columns `status`, `maximum_height`, and `minimum_height` are spotted.

There seems to be an useless column named `Unnamed: 0` too.

8 Data Cleaning

Now that we've gained some insights, we can start cleaning and manipulating the datasets.

For organization's sake, we'll go in the same order as the preliminar exploration: - Rasgos-CL species dataset. - Rasgos-CL geographical dataset. - Herbario Digital's dataset.

8.1 Rasgos-CL species

As we aim to know how studied the species are, which means we can reduce the size of our dataset.

[ ]:
```
# Reducing dataset to useful columns only.
ordered_traits = traits_df[["accepted_species", "traitName", "obs"]]

# Renaming columns for better readability
ordered_traits.rename(columns={"accepted_species": "specie", "traitName":␣
 ↪"trait_name"}, inplace=True)
```

We want to know how many observations a specie has got, so we can group our dataset by specie and sum their observations:

[ ]:
```
observed_species = ordered_traits.groupby("specie").agg({"obs": ["sum"]})
observed_species.columns = ["total_observations"]
observed_species.head()
```

[ ]:

	total_observations
specie	
Acrisione cymosa	27.0
Acrisione denticulata	49.0
Adenopeltis serrata	39.0
Adesmia aphylla	18.0
Adesmia argentea	25.0

8.2 Rasgos-CL geography

We need to translate encoded regions in order to improve its readability and enable the combination with Herbario Digital's dataset.

Also, we'll rename the columns with the same purposes stated above.

[ ]:
```
# Translate regions
new_regions = {
    'AIS': 'Aysén',
    'ANT': 'Antofagasta',
    'ARA': 'Araucanía',
    'ATA': 'Atacama',
    'AYP': 'Arica y Parinacota',
    'BIO': 'Bío-Bío',
    'COQ': 'Coquimbo',
    'IPA': 'Isla de Pascua',
    'JFE': 'Juan Fernández',
    'LBO': 'Libertador Bernardo O\'Higgins',
    'LLA': 'Los Lagos',
    'LRI': 'Los Ríos',
    'MAG': 'Magallanes',
    'MAU': 'Maule',
    'NUB': 'Ñuble',
    'RME': 'Metropolitana',
    'TAR': 'Tarapacá',
    'VAL': 'Valparaíso'
}
geo_df.replace(to_replace=new_regions, inplace=True)

# Rename columns
geo_df.columns = ["specie", "location", "is_present"]

# Check changes made
geo_df.head()
```

[ ]:
```
              specie            location  is_present
0  Acrisione cymosa               Aysén           1
1  Acrisione cymosa         Antofagasta           0
2  Acrisione cymosa           Araucanía           1
3  Acrisione cymosa             Atacama           0
4  Acrisione cymosa  Arica y Parinacota           0
```

8.3 Herbario Digital

As we mentioned during exploration, there's an unecessary column which should be dropped.

[ ]:
```
clean_herbario_df = herbario_species.drop(labels=["Unnamed: 0"], axis=1)
clean_herbario_df.head()
```

```
[ ]:      id                        scientific_name              habit   status  \
     0  1583                      Acrisione cymosa  Bush or Small tree  Endemic
     1  5567                 Acrisione denticulata                Bush   Native
     2  1585  Acrisione denticulata var. pilota                 Bush  Endemic
     3  3523                  Adenopeltis serrata                 Bush  Endemic
     4  3587                      Adesmia aphylla                 Bush  Endemic

       conservation_state  maximum_height  minimum_height  Araucanía  Maule  \
     0  Not Evaluated (NE)          1100.0             0.0          1      0
     1  Not Evaluated (NE)             NaN             NaN          0      0
     2  Not Evaluated (NE)          1200.0             0.0          1      1
     3  Not Evaluated (NE)           900.0             5.0          0      1
     4  Not Evaluated (NE)          3000.0          1800.0          0      0

        Atacama  …  Libertador Bernardo O'Higgins  Arica y Parinacota  Los Ríos  \
     0        0  …                              0                   0         1
     1        0  …                              0                   0         0
     2        0  …                              1                   0         0
     3        0  …                              1                   0         0
     4        1  …                              0                   0         0

       Ñuble  Coquimbo  Los Lagos  Magallanes  Bío-Bío  Valparaíso  Aysén
     0      0         0          1           0        1           1      1
     1      0         0          0           0        0           0      0
     2      1         1          1           0        1           1      0
     3      1         1          0           0        1           1      0
     4      0         1          0           0        0           0      0

     [5 rows x 24 columns]
```

9 Datasets merging

Now that our datasets look clean enough, we can start joining them to start analysing the whole information.

First, let's join Rasgos-CL datasets between them:

9.1 Joining Rasgos-CL datasets

First, we want to create one column per location so we can easily join and aggregate tabular data.

```
[ ]: location_dataframes = list()

     # Group by `specie` and keep `location` and `is_present` columns.
     grouped = geo_df.groupby("specie")[["location", "is_present"]]

     # For every specie, create a dataframe with its location as index.
```

```python
# Then, transpose that dataframe (columns become indices)
# Finally, collect those dataframes into `location_dataframes`
for specie_name in grouped.groups.keys():
    specie_df = grouped.get_group(specie_name).set_index("location").T
    specie_df.index = [specie_name]
    specie_df.columns.names = [""]
    location_dataframes.append(specie_df)

# Concatenate collected dataframes to form a complete dataset of located species
# As if locations were one-hot encoded.
located_species_df = pd.concat(location_dataframes)

located_species_df.head()
```

[ ]:

```
                        Aysén  Antofagasta  Araucanía  Atacama  \
Acrisione cymosa            1            0          1        0
Acrisione denticulata       0            0          1        0
Adenopeltis serrata         0            0          0        0
Adesmia aphylla             0            0          0        1
Adesmia argentea            0            0          0        1

                        Arica y Parinacota  Bío-Bío  Coquimbo  Isla de Pascua  \
Acrisione cymosa                         0        1         0             0.0
Acrisione denticulata                    0        1         1             0.0
Adenopeltis serrata                      0        1         1             0.0
Adesmia aphylla                          0        0         1             0.0
Adesmia argentea                         0        0         1             0.0

                        Juan Fernández  Libertador Bernardo O'Higgins  \
Acrisione cymosa                   0.0                              0
Acrisione denticulata              0.0                              1
Adenopeltis serrata                0.0                              1
Adesmia aphylla                    0.0                              0
Adesmia argentea                   0.0                              0

                        Los Lagos  Los Ríos  Magallanes  Maule  Ñuble  \
Acrisione cymosa                1         1           0      0      0
Acrisione denticulata           1         1           0      1      1
Adenopeltis serrata             0         0           0      1      1
Adesmia aphylla                 0         0           0      0      0
Adesmia argentea                0         0           0      0      0

                        Metropolitana  Tarapacá  Valparaíso
Acrisione cymosa                    0         0           1
Acrisione denticulata               1         0           1
Adenopeltis serrata                 1         0           1
Adesmia aphylla                     0         0           0
```

```
Adesmia argentea                    0        1           1
```

With the transformed table, we can clean it a little bit more by filling `NaN` values and casting floats to integers.

```python
[ ]: # Turn `NaN` into zeroes and cast into integers
     located_species_df.fillna(0, inplace=True)
     located_species_corrected_df = located_species_df.astype('int64')
     located_species_corrected_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 718 entries, Acrisione cymosa to Weinmannia trichosperma
Data columns (total 18 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Aysén                          718 non-null    int64
 1   Antofagasta                    718 non-null    int64
 2   Araucanía                      718 non-null    int64
 3   Atacama                        718 non-null    int64
 4   Arica y Parinacota             718 non-null    int64
 5   Bío-Bío                        718 non-null    int64
 6   Coquimbo                       718 non-null    int64
 7   Isla de Pascua                 718 non-null    int64
 8   Juan Fernández                 718 non-null    int64
 9   Libertador Bernardo O'Higgins  718 non-null    int64
 10  Los Lagos                      718 non-null    int64
 11  Los Ríos                       718 non-null    int64
 12  Magallanes                     718 non-null    int64
 13  Maule                          718 non-null    int64
 14  Ñuble                          718 non-null    int64
 15  Metropolitana                  718 non-null    int64
 16  Tarapacá                       718 non-null    int64
 17  Valparaíso                     718 non-null    int64
dtypes: int64(18)
memory usage: 106.6+ KB
```

And finally, we can merge the Rasgos-CL datasets together by their `specie` index:

```python
[ ]: observed_located_species = pd.merge(observed_species,
      ↪located_species_corrected_df, left_index=True, right_index=True)
     observed_located_species.head()
```

```
[ ]:                          total_observations  Aysén  Antofagasta  Araucanía  \
     specie
     Acrisione cymosa                       27.0      1            0          1
     Acrisione denticulata                  49.0      0            0          1
     Adenopeltis serrata                    39.0      0            0          0
     Adesmia aphylla                        18.0      0            0          0
```

```
Adesmia argentea                     25.0      0            0          0

                       Atacama  Arica y Parinacota  Bío-Bío  Coquimbo  \
specie
Acrisione cymosa             0                   0        1         0
Acrisione denticulata        0                   0        1         1
Adenopeltis serrata          0                   0        1         1
Adesmia aphylla              1                   0        0         1
Adesmia argentea             1                   0        0         1

                       Isla de Pascua  Juan Fernández  \
specie
Acrisione cymosa                    0               0
Acrisione denticulata               0               0
Adenopeltis serrata                 0               0
Adesmia aphylla                     0               0
Adesmia argentea                    0               0

                       Libertador Bernardo O'Higgins  Los Lagos  Los Ríos  \
specie
Acrisione cymosa                                   0          1         1
Acrisione denticulata                              1          1         1
Adenopeltis serrata                                1          0         0
Adesmia aphylla                                    0          0         0
Adesmia argentea                                   0          0         0

                       Magallanes  Maule  Ñuble  Metropolitana  Tarapacá  \
specie
Acrisione cymosa                0      0      0              0         0
Acrisione denticulata           0      1      1              1         0
Adenopeltis serrata             0      1      1              1         0
Adesmia aphylla                 0      0      0              0         0
Adesmia argentea                0      0      0              0         1

                       Valparaíso
specie
Acrisione cymosa                1
Acrisione denticulata           1
Adenopeltis serrata             1
Adesmia aphylla                 0
Adesmia argentea                1
```

9.2 Joining Herbario Digital dataset

Finally, we can join the Herbario Digital dataset to our located observations dataframe:

```
[ ]: # Merge on species' scientific names
merged = pd.merge(observed_located_species, herbario_species, left_index=True,␣
 ↪right_on="scientific_name", how="outer", suffixes=("", "_herbario"))
merged.head()
```

```
[ ]:     total_observations  Aysén  Antofagasta  Araucanía  Atacama  \
0.0                27.0    1.0          0.0        1.0      0.0
1.0                49.0    0.0          0.0        1.0      0.0
2.0                 NaN    NaN          NaN        NaN      NaN
3.0                39.0    0.0          0.0        0.0      0.0
4.0                18.0    0.0          0.0        0.0      1.0

     Arica y Parinacota  Bío-Bío  Coquimbo  Isla de Pascua  Juan Fernández  \
0.0                 0.0      1.0       0.0             0.0             0.0
1.0                 0.0      1.0       1.0             0.0             0.0
2.0                 NaN      NaN       NaN             NaN             NaN
3.0                 0.0      1.0       1.0             0.0             0.0
4.0                 0.0      0.0       1.0             0.0             0.0

     …  Libertador Bernardo O'Higgins_herbario  Arica y Parinacota_herbario  \
0.0  …                                     0.0                          0.0
1.0  …                                     0.0                          0.0
2.0  …                                     1.0                          0.0
3.0  …                                     1.0                          0.0
4.0  …                                     0.0                          0.0

     Los Ríos_herbario  Ñuble_herbario  Coquimbo_herbario  Los Lagos_herbario  \
0.0                1.0             0.0                0.0                 1.0
1.0                0.0             0.0                0.0                 0.0
2.0                0.0             1.0                1.0                 1.0
3.0                0.0             1.0                1.0                 0.0
4.0                0.0             0.0                1.0                 0.0

     Magallanes_herbario  Bío-Bío_herbario  Valparaíso_herbario  \
0.0                  0.0               1.0                  1.0
1.0                  0.0               0.0                  0.0
2.0                  0.0               1.0                  1.0
3.0                  0.0               1.0                  1.0
4.0                  0.0               0.0                  0.0

     Aysén_herbario
0.0             1.0
1.0             0.0
2.0             0.0
3.0             0.0
4.0             0.0
```

```
[5 rows x 44 columns]
```

There's a problem, though.

We've got repeated columns which do not hold the same information! e.g. according to Herbario Digital, "Acrisione denticulata" is absent in "Coquimbo", but it's present according to Rasgos-CL!

We must fix this by unifying the location data.

```
[ ]: # Get all possible regions
region_columns = [region.split("_")[0] for region in merged.columns if␣
 ↪"_herbario" in region]

# Create a new dataframe to hold the unified data.
unified_regions = merged
for region in region_columns:
    # For every region, set it's value if whether Rasgos-CL or Herbario Digital␣
 ↪has valid data. If not, set it to 0.
    unified_regions[region] = unified_regions.apply(lambda row:␣
 ↪row[f'{region}_herbario'] if (row[region]==0 or np.isnan(row[region])) and␣
 ↪not np.isnan(row[f'{region}_herbario']) else row[region], axis=1)

unified_regions.head()
```

```
[ ]:      total_observations  Aysén  Antofagasta  Araucanía  Atacama  \
0.0                    27.0    1.0          0.0        1.0      0.0
1.0                    49.0    0.0          0.0        1.0      0.0
2.0                     NaN    0.0          0.0        1.0      0.0
3.0                    39.0    0.0          0.0        0.0      0.0
4.0                    18.0    0.0          0.0        0.0      1.0

     Arica y Parinacota  Bío-Bío  Coquimbo  Isla de Pascua  Juan Fernández  \
0.0                 0.0      1.0       0.0             0.0             0.0
1.0                 0.0      1.0       1.0             0.0             0.0
2.0                 0.0      1.0       1.0             NaN             0.0
3.0                 0.0      1.0       1.0             0.0             0.0
4.0                 0.0      0.0       1.0             0.0             0.0

     …  Libertador Bernardo O'Higgins_herbario  Arica y Parinacota_herbario  \
0.0  …                                     0.0                          0.0
1.0  …                                     0.0                          0.0
2.0  …                                     1.0                          0.0
3.0  …                                     1.0                          0.0
4.0  …                                     0.0                          0.0

     Los Ríos_herbario  Ñuble_herbario  Coquimbo_herbario  Los Lagos_herbario  \
0.0                1.0             0.0                0.0                 1.0
1.0                0.0             0.0                0.0                 0.0
```

```
2.0                    0.0              1.0                 1.0                   1.0
3.0                    0.0              1.0                 1.0                   0.0
4.0                    0.0              0.0                 1.0                   0.0

     Magallanes_herbario  Bío-Bío_herbario  Valparaíso_herbario  \
0.0                  0.0               1.0                  1.0
1.0                  0.0               0.0                  0.0
2.0                  0.0               1.0                  1.0
3.0                  0.0               1.0                  1.0
4.0                  0.0               0.0                  0.0

     Aysén_herbario
0.0             1.0
1.0             0.0
2.0             0.0
3.0             0.0
4.0             0.0

[5 rows x 44 columns]
```

Now, let's simplify our dataset by selecting a handful of useful columns and adding a "location score", which is a measure of how distributed a specie is, e.i. in how many locations it's present:

[ ]:
```
simplified_columns = ["scientific_name", "total_observations", "habit",␣
  ↪"status", "conservation_state", "maximum_height", "minimum_height", "Isla de␣
  ↪Pascua"]
simplified_columns.extend(region_columns)

simplified_df = unified_regions.loc[:, simplified_columns]

# Add location score by showing total region presence
simplified_df["location_score"] = simplified_df.loc[:, region_columns].
  ↪sum(axis=1)
simplified_df.head()
```

[ ]:
```
                          scientific_name  total_observations  \
0.0                      Acrisione cymosa                27.0
1.0                 Acrisione denticulata                49.0
2.0  Acrisione denticulata var. pilota                  NaN
3.0                   Adenopeltis serrata                39.0
4.0                       Adesmia aphylla                18.0

                  habit   status  conservation_state  maximum_height  \
0.0  Bush or Small tree  Endemic  Not Evaluated (NE)          1100.0
1.0                Bush   Native  Not Evaluated (NE)             NaN
2.0                Bush  Endemic  Not Evaluated (NE)          1200.0
3.0                Bush  Endemic  Not Evaluated (NE)           900.0
```

```
4.0                Bush  Endemic  Not Evaluated (NE)         3000.0

     minimum_height  Isla de Pascua  Araucanía  Maule  …  \
0.0             0.0             0.0        1.0    0.0  …
1.0             NaN             0.0        1.0    1.0  …
2.0             0.0             NaN        1.0    1.0  …
3.0             5.0             0.0        0.0    1.0  …
4.0          1800.0             0.0        0.0    0.0  …

     Arica y Parinacota  Los Ríos  Ñuble  Coquimbo  Los Lagos  Magallanes  \
0.0                 0.0       1.0    0.0       0.0        1.0         0.0
1.0                 0.0       1.0    1.0       1.0        1.0         0.0
2.0                 0.0       0.0    1.0       1.0        1.0         0.0
3.0                 0.0       0.0    1.0       1.0        0.0         0.0
4.0                 0.0       0.0    0.0       1.0        0.0         0.0

     Bío-Bío  Valparaíso  Aysén  location_score
0.0      1.0         1.0    1.0             6.0
1.0      1.0         1.0    0.0            10.0
2.0      1.0         1.0    0.0             9.0
3.0      1.0         1.0    0.0             7.0
4.0      0.0         0.0    0.0             2.0

[5 rows x 26 columns]
```

Another problem is that there are species with `NaN` values for the `total_observations` column.

Those species are useless to us as we want to analyse the observations per specie, so we might as well drop those while replacing other `NaN` values:

[ ]:
```python
# Drop species with no registered observations (`NaN`)
clean_df = simplified_df.dropna(subset=["total_observations"])

# Create a dictionary used to replace `NaN` values
column_fillna = {region: 0 for region in region_columns}
column_fillna["total_observations"] = 0
column_fillna["conservation_state"] = "Not Applicable (N/A)"

clean_df.fillna(column_fillna, inplace=True)

# Remove Herbario's assigned index
clean_df = clean_df.reset_index().drop("index", axis=1)

# Show result, sorted by observations
clean_df.sort_values(by="total_observations", ascending=False).head()
```

[ ]:
```
       scientific_name  total_observations habit  status  \
478    Maytenus boaria               250.0  Tree  Native
```

```
228     Drimys winteri                230.0   NaN     NaN
566      Persea lingue                218.0  Tree  Native
713    Vachellia caven                207.0  Tree  Native
588  Prosopis tamarugo                189.0   NaN     NaN

       conservation_state  maximum_height  minimum_height  Isla de Pascua  \
478    Not Evaluated (NE)          4000.0             0.0             0.0
228  Not Applicable (N/A)             NaN             NaN             0.0
566    Least Concern (LC)           900.0             0.0             0.0
713    Not Evaluated (NE)          3000.0             0.0             0.0
588  Not Applicable (N/A)             NaN             NaN             0.0

     Araucanía  Maule  …  Arica y Parinacota  Los Ríos  Ñuble  Coquimbo  \
478        1.0    1.0  …                 0.0       1.0    1.0       1.0
228        1.0    1.0  …                 0.0       1.0    1.0       1.0
566        1.0    1.0  …                 0.0       1.0    1.0       0.0
713        1.0    1.0  …                 0.0       1.0    1.0       1.0
588        0.0    0.0  …                 1.0       0.0    0.0       0.0

     Los Lagos  Magallanes  Bío-Bío  Valparaíso  Aysén  location_score
478        1.0         1.0      1.0         1.0    1.0            13.0
228        1.0         1.0      1.0         1.0    1.0            12.0
566        1.0         0.0      1.0         1.0    0.0             9.0
713        0.0         0.0      1.0         1.0    0.0            10.0
588        0.0         0.0      0.0         0.0    0.0             3.0

[5 rows x 26 columns]
```

Note: `habit`, `status`, and `*_height` columns get to keep their `NaN` values as no obvious replacement was found.

10 Analysing and plotting the data

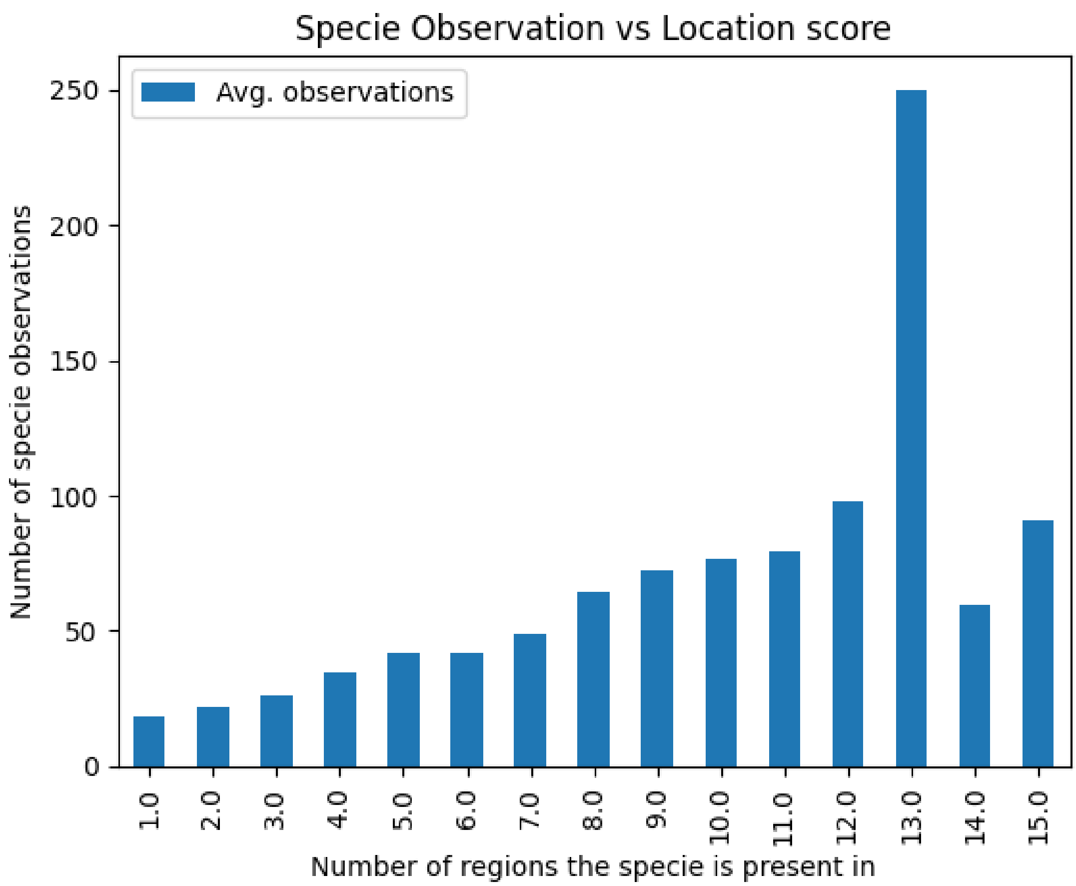
Now we get to plot different combinations of data attributes to get some deeper insights.

10.0.1 Specie observation by distribution

Explanation: Up to a point, the observations of a specie increases along with the number of locations it's distributed in. The wider the distribution, the higher number of observations.

[ ]:
```
obs_per_location = clean_df.loc[:, ["location_score", "total_observations"]].
 ↪groupby("location_score").mean().sort_values("location_score")

plot_bar(obs_per_location,
    title="Specie Observation vs Location score",
    xlabel="Number of regions the specie is present in",
    ylabel="Number of specie observations",
```

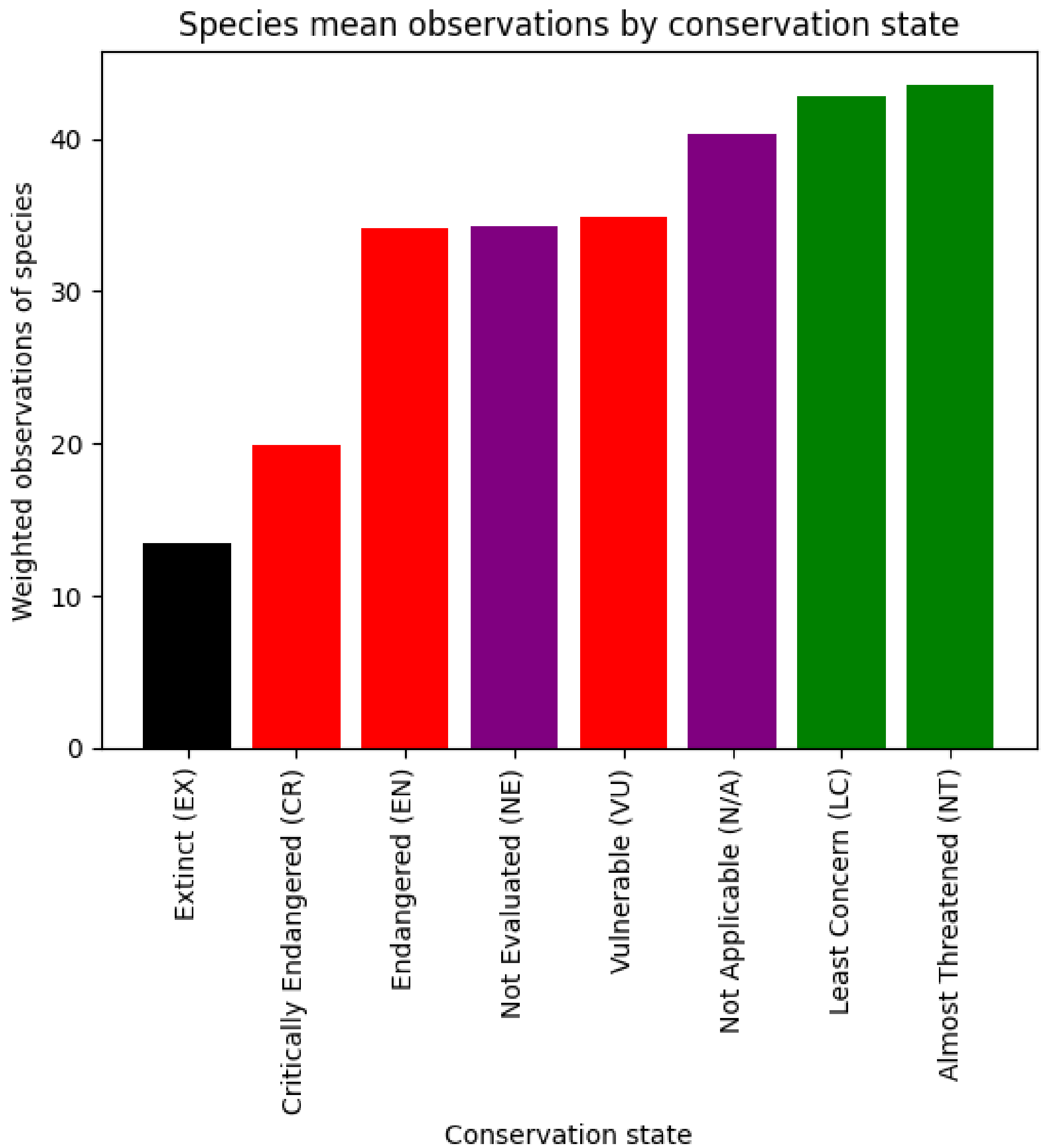
```
    legend=["Avg. observations"])
```



10.0.2 Specie obsrvation by consrvation state

Explanation: Species with lower risk of extinction tend to have more observations than endangered species, probably because of the smaller populations available in the wild.

```
[ ]: state_obs = clean_df.loc[:, ["total_observations", "conservation_state"]].
     ↪groupby("conservation_state").mean().sort_values("total_observations").
     ↪reset_index()

     plot_bar(
         x=state_obs.conservation_state, y=state_obs.total_observations,
         title="Species mean observations by conservation state",
         xlabel="Conservation state",
         ylabel="Weighted observations of species",
         color=["black", "red", "red", "purple", "red", "purple", "green", "green"]
     )
```



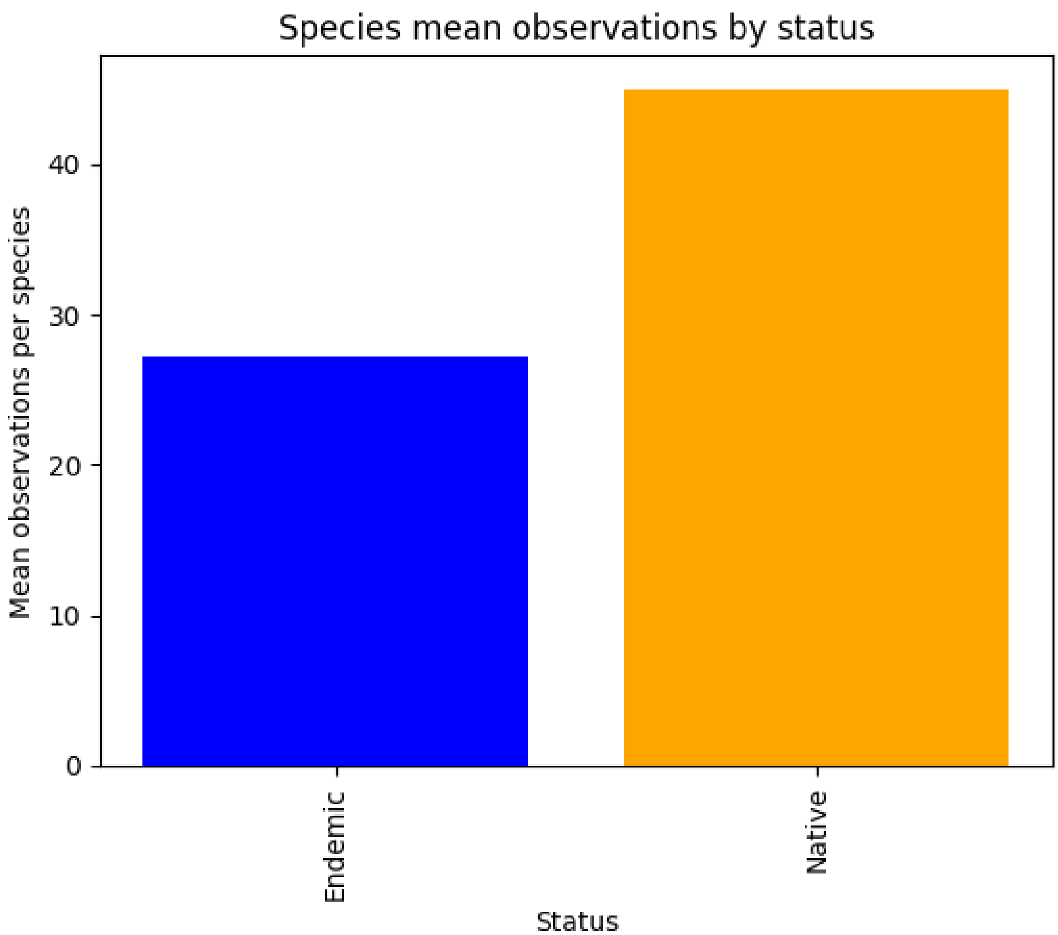
10.0.3 Species observation by specie status

Note: > Native species are those that have evolved and existed in a particular region for a long time, without any human intervention. On the other hand, endemic species are a subset of native species that are exclusively found in a specific geographic area and are not naturally found anywhere else in the world.

— https://thisvsthat.io/endemic-vs-native

Explanation: Similar to previous results, smaller or more isolated populations tend to be less observed.

```
[ ]: status_obs = clean_df.loc[:, ["total_observations", "status"]].
     ↪groupby("status").mean().sort_values("total_observations").reset_index()

     plot_bar(
         x=status_obs.status, y=status_obs.total_observations,
         title="Species mean observations by status",
         xlabel="Status",
         ylabel="Mean observations per species",
         color=["blue", "orange"]
     )
```



10.0.4 Species distribution by status

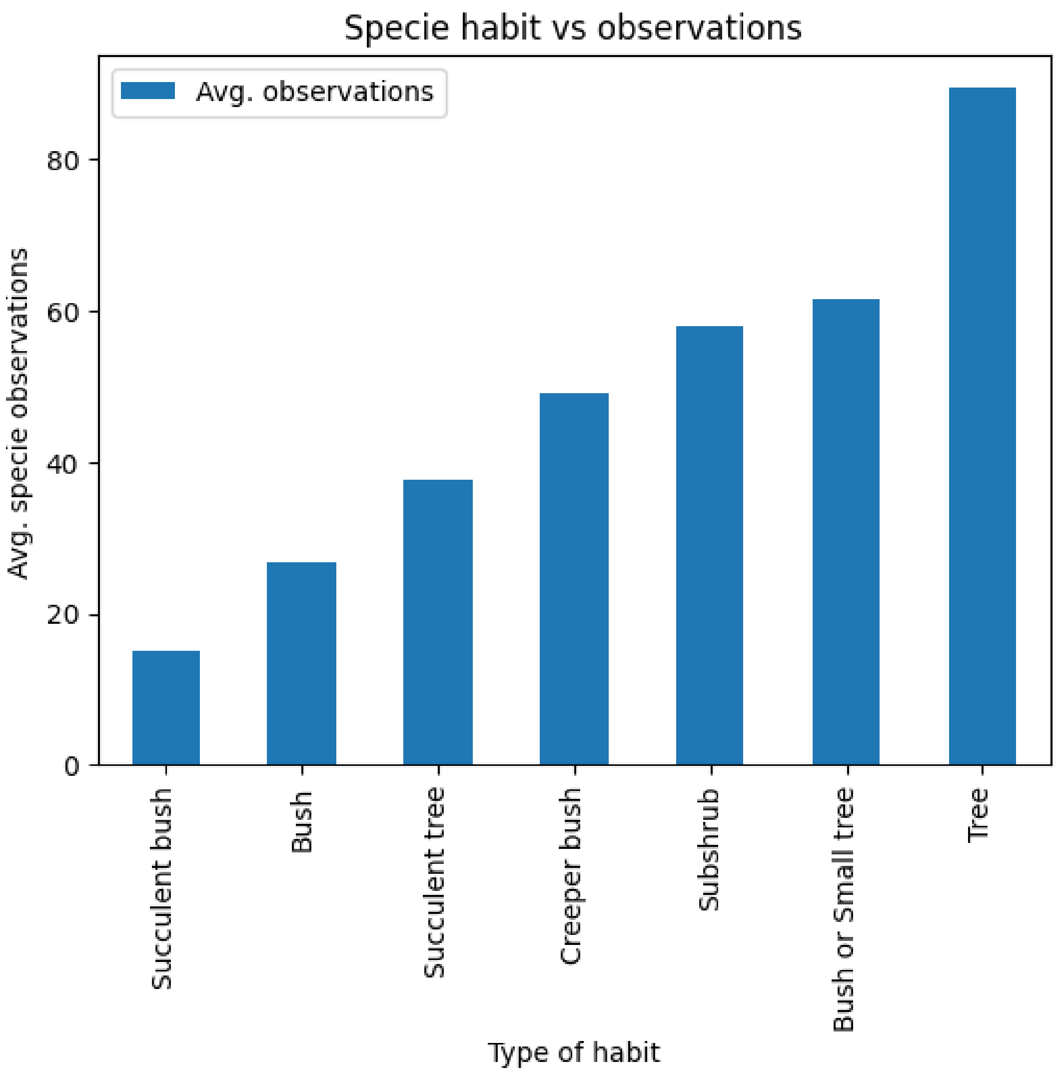
Explanation: By definition, an endemic specie might be harder to spot as it doesn't survive when placed in foreign habitats. This might be corroborated by looking at the location score vs the status of the species:

[ ]:
```
status_locations = clean_df.loc[:, ["status", "location_score"]].
 ↪groupby("status").mean().sort_values("location_score").reset_index()

plot_bar(
    x=status_locations.status, y=status_locations.location_score,
    title="Species mean locations by status",
    xlabel="Status",
    ylabel="Avg. locations per specie",
    color=["blue", "orange"]
)
```

10.0.5 Specie observations by grow habit:

Explanation: Different types of plants, defined here by its growth habit, have different number of average observations, showing that trees and small trees have been subject to more observations than different types of bushes or shrubs. This might be correlated to height data.

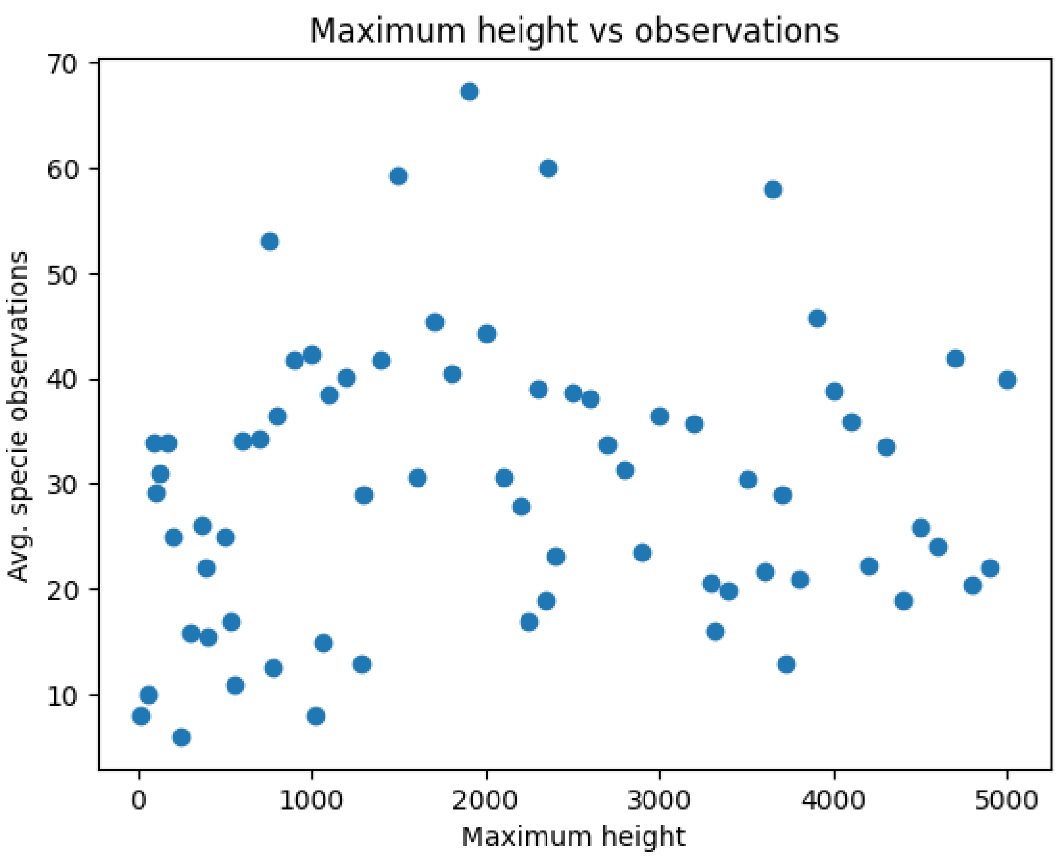
[ ]:
```
obs_per_habit = clean_df.loc[:, ["habit", "total_observations"]].
↪groupby("habit").mean().sort_values("total_observations")

plot_bar(
    data=obs_per_habit,
    title="Specie habit vs observations",
    xlabel="Type of habit",
    ylabel="Avg. specie observations",
    legend=["Avg. observations"]
)
```



10.0.6 Species observations by maximum height

Explanation: Any trend here is probably due to randomness rather than a existent correlation. We might expect the same for the minimum height attribute.

[ ]:
```
obs_per_max_height = clean_df.loc[:, ["maximum_height", "total_observations"]].
↪groupby("maximum_height").mean().sort_values("maximum_height").reset_index()

plot_scatter(x=obs_per_max_height.maximum_height, y=obs_per_max_height.
↪total_observations,title="Maximum height vs observations",
    xlabel="Maximum height",
    ylabel="Avg. specie observations")
```

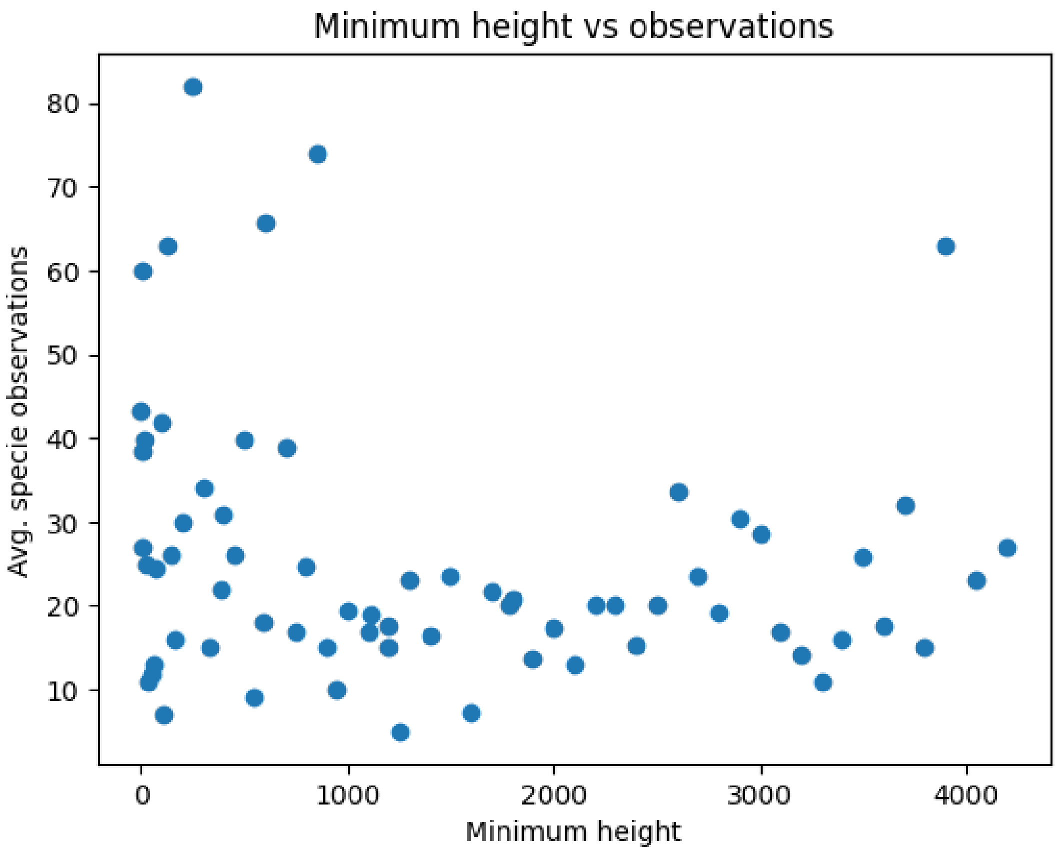


10.0.7 Species observations by minimum height

Explanation: As expected, no real trend here.

[ ]:
```
obs_per_min_height = clean_df.loc[:, ["minimum_height", "total_observations"]].
↪groupby("minimum_height").mean().sort_values("minimum_height").reset_index()

plot_scatter(
```

```
    x=obs_per_min_height.minimum_height, y=obs_per_min_height.
 ↪total_observations,
    title="Minimum height vs observations",
    xlabel="Minimum height",
    ylabel="Avg. specie observations"
)
```



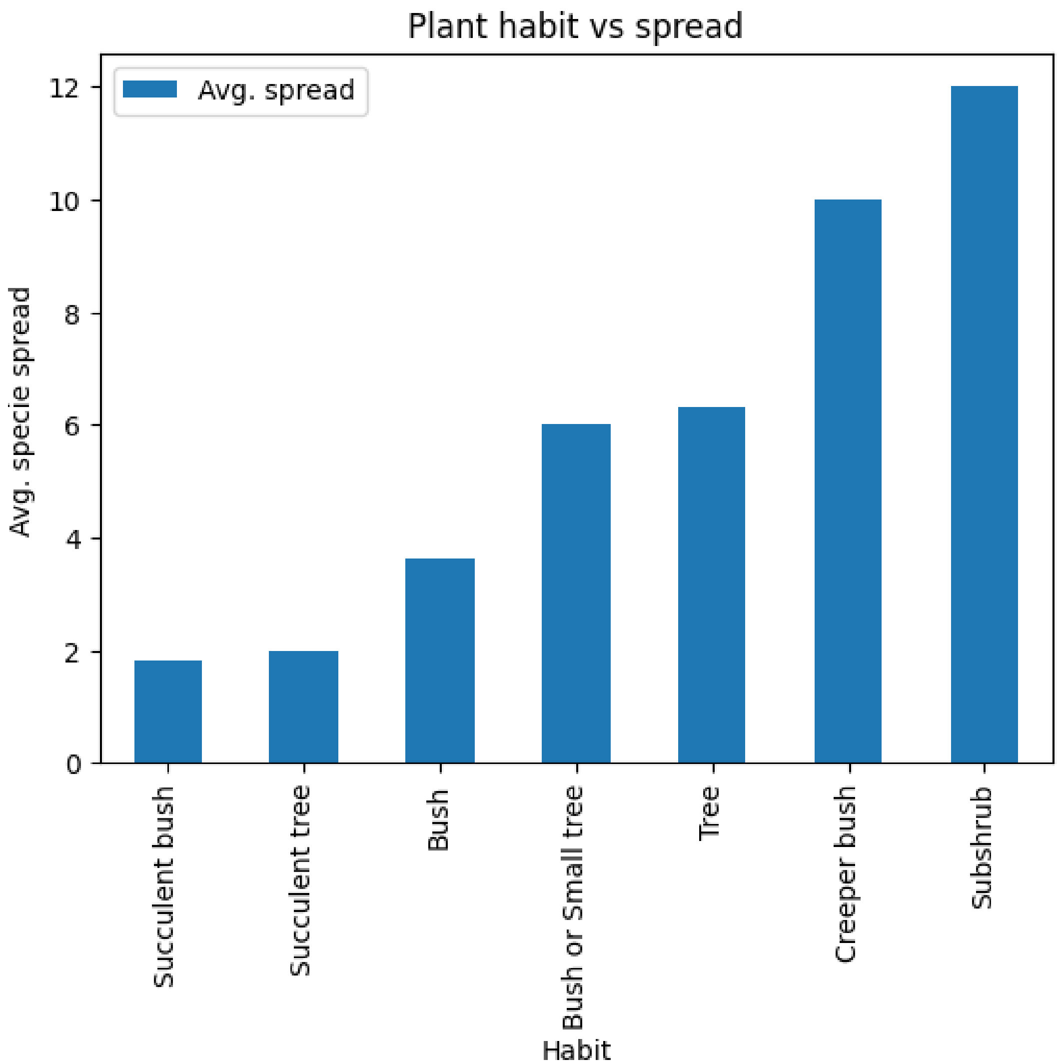
Height looks like doesn't correlate at all with the level of study of the species.

10.0.8 Species distribution by habit

Explanation: Even though trees and small trees have more observations in average, they aren't the most spreaded habits.

[ ]:
```
locations_per_habit = clean_df.loc[:, ["habit", "location_score"]].
 ↪groupby("habit").mean().sort_values("location_score")

plot_bar(
    data=locations_per_habit,
    title="Plant habit vs spread",
```

```
    xlabel="Habit",
    ylabel="Avg. specie spread",
    legend=["Avg. spread"]
)
```

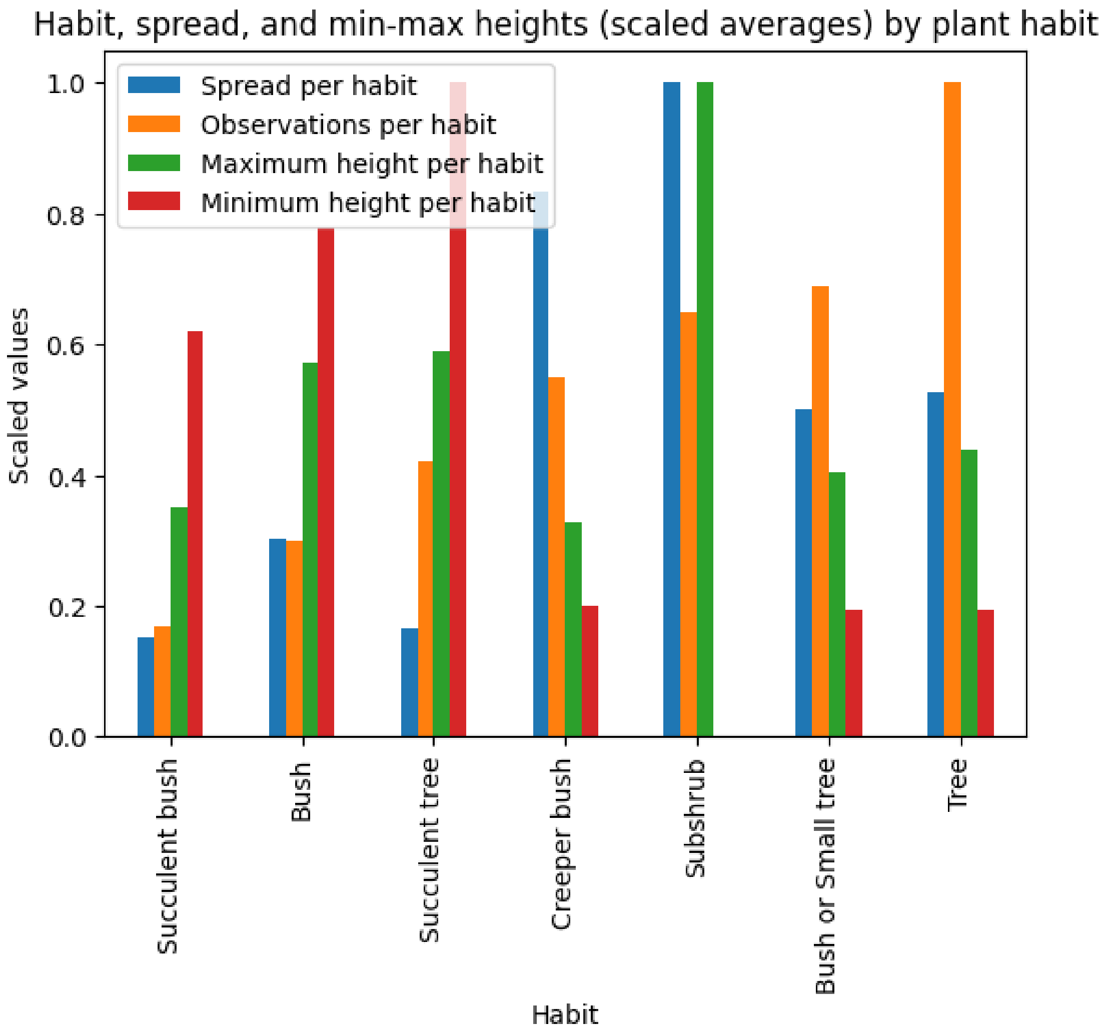


10.0.9 Distribution, observations, and maximum and minimum heights by plant habit

Explanation: As expected, the only variables that seem to really correlate with the number of observations are the distribution and habit of the specie. Height data looks like noise.

[ ]:
```
max_height_per_habit = clean_df.loc[:, ["habit", "maximum_height"]].
  ↪groupby("habit").mean().sort_values("maximum_height")
min_height_per_habit = clean_df.loc[:, ["habit", "minimum_height"]].
  ↪groupby("habit").mean().sort_values("minimum_height")
```

```python
# Create a new dataframe with the mean values for each feature of interest␣
↪regarding habit and distribution
habit_comparison = pd.concat(
    [
        locations_per_habit.T.mean().rename("Spread per habit"),
        obs_per_habit.T.mean().rename("Observations per habit"),
        max_height_per_habit.T.mean().rename("Maximum height per habit"),
        min_height_per_habit.T.mean().rename("Minimum height per habit")
    ],
    axis=1
)

# Scale data to aid visualization

scaler = MaxAbsScaler()
habit_scaled = pd.DataFrame(scaler.fit_transform(habit_comparison),␣
↪columns=habit_comparison.columns, index=habit_comparison.index).
↪sort_values("Observations per habit")

plot_bar(
    data=habit_scaled,
    title="Habit, spread, and min-max heights (scaled averages) by plant habit",
    xlabel="Habit",
    ylabel="Scaled values",
)
```



10.1 Numeric correlations

Now, we can also analyse specific correlations among different features.

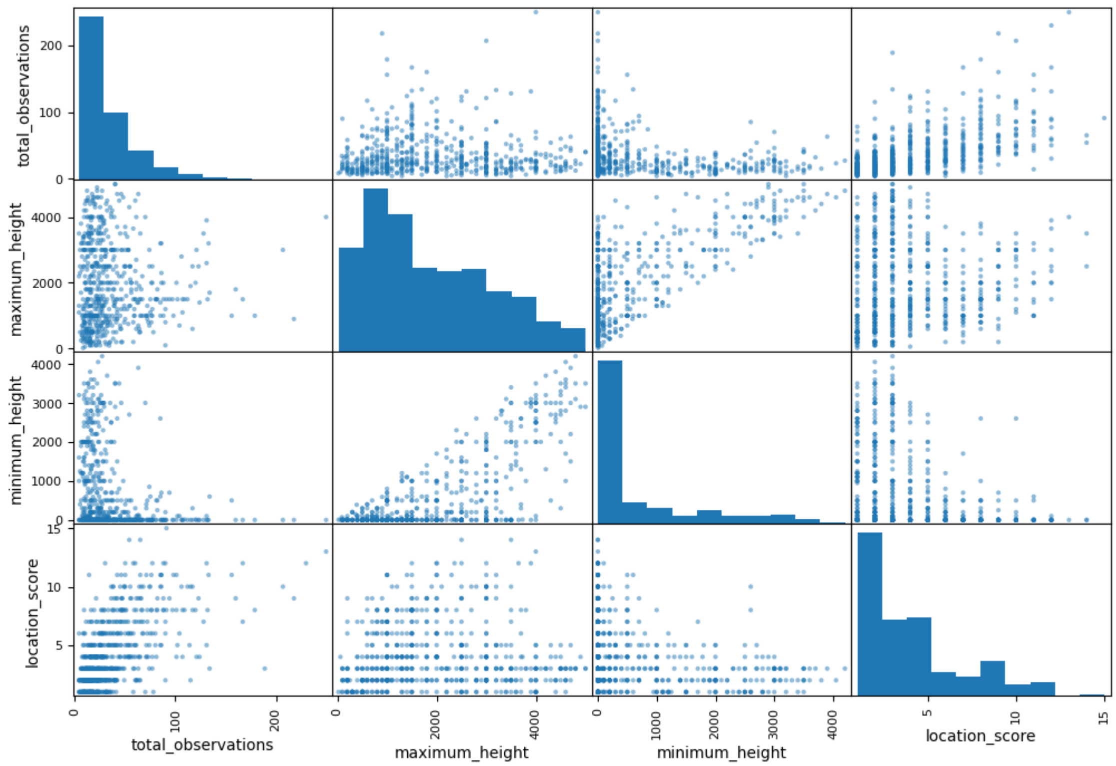
[ ]:
```
num_cols = ['total_observations', 'maximum_height', 'minimum_height',␣
 ↪'location_score']
cat_cols = ['habit', 'status', 'conservation_state']
correlation_matrix = clean_df.loc[:, num_cols].corr()
correlation_matrix["total_observations"].sort_values(ascending=False)
```

[ ]:
```
total_observations     1.000000
location_score         0.597608
maximum_height        -0.019107
minimum_height        -0.244236
Name: total_observations, dtype: float64
```

Confirmed, the distribution of species has a correlation index of 0.597 with the observations, which is pretty high!

Plotting this correlation would look like this:

```
[ ]: from pandas.plotting import scatter_matrix

     scatter_matrix(clean_df[num_cols], figsize=(12, 8))
     plt.show()
```



You can visualize the correlation by looking at the top-right scatter plot.

Regarding the categorical features, we can encode their values and perform a similar analysis:

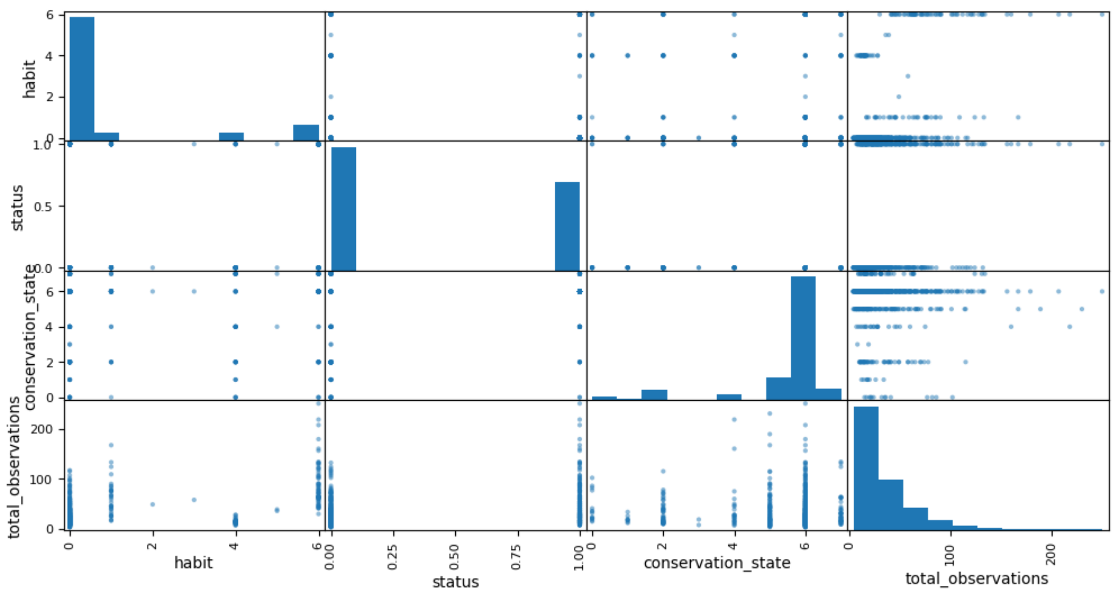
```
[ ]: categoric_df = clean_df[cat_cols]

     ordinal_encoder = OrdinalEncoder()
     categoric_encoded = pd.DataFrame(ordinal_encoder.fit_transform(categoric_df),
      ↪columns=cat_cols)
     categoric_encoded["total_observations"] = clean_df.reset_index().drop("index",
      ↪axis=1)["total_observations"]


     correlation_matrix = categoric_encoded.corr()
```

```
correlation_matrix["total_observations"].sort_values(ascending=False)
```

[ ]:
```
total_observations     1.000000
habit                  0.506989
status                 0.284135
conservation_state    -0.017514
Name: total_observations, dtype: float64
```

Here, looks like the encoded values of the habit feature is positively correlated with the number of observations!

[ ]:
```
scatter_matrix(categoric_encoded, figsize=(12, 6))
plt.show()
```



To understand these numbers, let's check their categorical values:

[ ]:
```
habit_cats = ordinal_encoder.categories_[0]
{idx: habit_cats[idx] for idx in range(len(habit_cats))}
```

[ ]:
```
{0: 'Bush',
 1: 'Bush or Small tree',
 2: 'Creeper bush',
 3: 'Subshrub',
 4: 'Succulent bush',
 5: 'Succulent tree',
 6: 'Tree',
 7: nan}
```

10.2 Observations by location

Now, something interesting is to see whether researchers might have "favorite" locations to register data on flora. This is an important source of bias in the registered data.

[ ]:
```
# Separate location columns
non_location_cols = ['scientific_name', 'total_observations', 'habit',␣
↪'status', 'conservation_state', 'maximum_height', 'minimum_height',␣
↪'location_score']
location_cols = [location for location in clean_df.columns if location not in␣
↪non_location_cols]

# Get the sum and average observations per location
located_obs = [
    [
        location,
        int(clean_df.loc[clean_df[location] == 1, "total_observations"].sum()),
        float(clean_df.loc[clean_df[location] == 1, "total_observations"].
↪mean())
    ]
    for location in location_cols]

sorted_by_total = sorted(located_obs, key=lambda location: location[1],␣
↪reverse=True)
sorted_by_average = sorted(located_obs, key=lambda location: location[2],␣
↪reverse=True)
```

[ ]:
```
sorted_by_total
```

[ ]:
```
[['Maule', 12836, 54.16033755274262],
 ['Coquimbo', 12598, 40.12101910828026],
 ['Valparaíso', 12021, 47.89243027888446],
 ['Bío-Bío', 11825, 58.251231527093594],
 ['Metropolitana', 11493, 48.90638297872341],
 ['Ñuble', 11293, 60.715053763440864],
 ["Libertador Bernardo O'Higgins", 10898, 54.49],
 ['Araucanía', 10744, 61.394285714285715],
 ['Los Ríos', 8849, 66.03731343283582],
 ['Los Lagos', 8440, 63.45864661654135],
 ['Atacama', 7790, 31.03585657370518],
 ['Aysén', 5578, 64.86046511627907],
 ['Antofagasta', 5509, 28.396907216494846],
 ['Tarapacá', 4563, 32.59285714285714],
 ['Arica y Parinacota', 3840, 35.55555555555556],
 ['Magallanes', 3591, 56.109375],
 ['Juan Fernández', 636, 70.66666666666667],
 ['Isla de Pascua', 0, nan]]
```

[ ]:
```
sorted_by_average
```

[ ]:
```
[['Isla de Pascua', 0, nan],
 ['Juan Fernández', 636, 70.66666666666667],
 ['Los Ríos', 8849, 66.03731343283582],
 ['Aysén', 5578, 64.86046511627907],
 ['Los Lagos', 8440, 63.45864661654135],
 ['Araucanía', 10744, 61.394285714285715],
 ['Ñuble', 11293, 60.715053763440864],
 ['Bío-Bío', 11825, 58.251231527093594],
 ['Magallanes', 3591, 56.109375],
 ["Libertador Bernardo O'Higgins", 10898, 54.49],
 ['Maule', 12836, 54.16033755274262],
 ['Metropolitana', 11493, 48.90638297872341],
 ['Valparaíso', 12021, 47.89243027888446],
 ['Coquimbo', 12598, 40.12101910828026],
 ['Arica y Parinacota', 3840, 35.55555555555556],
 ['Tarapacá', 4563, 32.59285714285714],
 ['Atacama', 7790, 31.03585657370518],
 ['Antofagasta', 5509, 28.396907216494846]]
```

Note: *Omitting "Isla de Pascua" as no specie appears to have that location.*

We can appreciate that the absolute and relative observations predominate in southern regions, specially below "*Región Metropolitana*".

This makes sense as a Chilean citizen, as those regions are specially humid, are rich in flora species, have big forest areas, and are specially touristic!

Whereas northern regions are dominated by deserts and are also less appealing for tourists.

11 Conclusions

Summarizing all the insights obtained through data correlation and plotting, we can draw certain conclusions:

1. Distribution, that is, the number of locations in which a species is found, significantly affects the probability that it will be better studied.
2. Species less sensitive to habitat change, that is, native species rather than endemic species, are easier to study since they comply with the previous point.
3. The height at which certain species develop does not seem to influence their study.
4. It seems that regions with greater tourism or those known to be richer in flora and forested areas receive more researchers or allow more exhaustive observation of flora species.
5. Less vulnerable species are more studied, again this is related to point 1 as it is understood that a less threatened species would have larger populations and, therefore, would be easier to find and observe.
6. Certain types of plants, such as trees, are more studied.

In other words, the ease of finding species is the determining factor when studying them in depth.

Regional forms of government could carry out initiatives to facilitate the study of flora in their localities to reduce the gap in the study of flora species in Chile and, thus, reduce biases in the available data repositories.

---

12 References:

- **Rasgos-CL repository:** https://github.com/dylancraven/Rasgos-CL?tab=readme-ov-file
- **Herbario Digital:** https://herbariodigital.cl/
- **Herbario Digital public API documentation:** https://api.herbariodigital.cl/swagger-ui/
- **Working with groups:** https://realpython.com/pandas-groupby/#example-1-us-congress-dataset
- **IUCN Red List of Threatened Species:** https://en.wikipedia.org/wiki/Conservation_status
- **Endemic vs native species:** https://thisvsthat.io/endemic-vs-native

13 Acknowledgments:

- **Herbario Digital:** IEB, CONC & ULS (2024) Herbario Digital. Publicado en Internet; https://www.herbariodigital.cl. Acceso en: 03 jul 2024

- **Rasgos-CL:** Alfaro, E., Pérez-Tello, V., Acevedo, M., Ovalle, J., Segovia, R., & Craven, D. (2023). Rasgos-CL: A functional trait database of Chilean woody plants. Global Ecology and Biogeography, 32, 2072–2084. https://doi.org/10.1111/geb.13755