# Impact of forest treatment on species

We assessed the impact on the number of species present in tropical and temperate forests after application of a pesticide (treatment), vs forests which were left untreated (controls).

Mean (sd) number of species, for untreated: 200.7 9.98,

Mean (sd) number of species, for treated: 199.20 9.81

We found a significant difference in the number of species between the groups. The treated forest showed a reduced number of species (P<0.001, Cohen’s *d*: 0.15 –Figure 1), suggesting that the treatment negatively impacts the biodiversity in the forest. This difference was slight however, given the reduced Cohen’s *d,* indicating that further testing is needed to assess the impact of the treatment.

Figure 1:

A blue and orange lines with words

Description automatically generated

Code snippet – available: <https://github.com/diffuze/imperial_test>

# -\*- coding: utf-8 -\*-

"""

Created on Sat Sep 2 12:44:05 2023

@author: medchaya

"""

import pandas as pd

import numpy as np

from scipy import stats

from matplotlib import pyplot as plt

np.random.seed(42)

# number of individuals

n\_treated = 1000

n\_untreated = 1000

mean\_species = 200

sd\_species = 10

# initial dataframe

## Distributions for the two groups

# treated forests slightly fewer species

num\_species\_t = pd.Series(np.round(np.random.normal(mean\_species-1,sd\_species,n\_treated)))

num\_species\_nt = pd.Series(np.round(np.random.normal(mean\_species,sd\_species,n\_untreated)))

type\_t = pd.Series(np.random.choice(('temperate','tropical'),n\_treated))

type\_nt = pd.Series(np.random.choice(('temperate','tropical'),n\_untreated))

## Full df

treated\_status = np.append(np.zeros(n\_treated), np.ones(n\_untreated)).astype(bool)

df = pd.DataFrame({'treated\_status':treated\_status,

'n\_species':pd.concat([num\_species\_nt,num\_species\_t]),

'type':pd.concat([type\_t,type\_nt])})

g\_untreated = df.loc[df['treated\_status']==False, 'n\_species']

g\_treated = df.loc[df['treated\_status']==True, 'n\_species']

## plot num species per group

plt.boxplot((g\_untreated,g\_treated))

for x,v in enumerate(zip((n\_untreated, n\_treated), [False,True])):

n = v[0]

status = v[1]

plt.scatter(pd.Series([x+1]\*n) + np.random.normal(0,.02,n),

df.loc[df['treated\_status']==status, 'n\_species'], alpha=0.5)

plt.xticks((1,2),['untreated','treated'])

# Compare differences in number of species:

t\_res = stats.ttest\_ind(g\_untreated,

g\_treated)

t\_res\_pval = t\_res[1]

print('P val: {}'.format(t\_res\_pval))

## effect size:

cohens\_d = (np.mean(g\_untreated) - np.mean(g\_treated)) / (np.sqrt((np.std(g\_untreated) \*\* 2 + np.std(g\_treated) \*\* 2) / 2))

print('Cohens D: {}'.format(cohens\_d))