Tourism and Chemical Water Pollution: A Global Analysis

Data Analysis Report

Report done by Thilina Abhisheka

Introduction

This report contains the python code of the analysis, results and the conclusion of the Tourism and Chemical Water Pollution research.

Vocabulary

- csw = coastal surface water
- cs = coastal sediment
- cb = coastal biota
- isw = inland surface water
- is = inland sediment
- ib = inland biota

Importing data and libraries

In [1]:
```
#import libraries

import pandas as pd
import seaborn as sns
from scipy.stats import shapiro
from scipy import stats
import numpy as np
from pingouin import kruskal
```

In [2]:
```
#import dataset

df_csw = pd.read_excel("datasets/coastal_surface_water.xlsx")
df_cs = pd.read_excel("datasets/coastal_sediment.xlsx")
df_cb = pd.read_excel("datasets/coastal_biota.xlsx")
df_isw = pd.read_excel("datasets/inland_surface_water.xlsx")
df_is = pd.read_excel("datasets/inland_sediment.xlsx")
df_ib = pd.read_excel("datasets/inland_biota.xlsx")
```

Data Cleaning

In [3]:
```
# Remove unit functions

def remove_µgL(data):
    unit = " µg/L"
    return data.strip(unit)

def remove_µgg1(data):
    unit = " µg/g"
    return data.strip(unit)

def remove_µgg2(data):
    unit = " µg/g"
    return data.strip(unit)
```

In [4]:
```
# Removing units

df_csw['Concentration'] = df_csw.Concentration.apply(remove_µgL)
df_cs['Concentration'] = df_cs.Concentration.apply(remove_µgg1)
df_cb['Concentration'] = df_cb.Concentration.apply(remove_µgg1)
df_isw['Concentration'] = df_isw.Concentration.apply(remove_µgL)
df_is['Concentration'] = df_is.Concentration.apply(remove_µgg1)
df_ib['Concentration'] = df_ib.Concentration.apply(remove_µgg2)
```

In [5]:
```
# print(df_csw)
# print(df_cs)
# print(df_cb)
# print(df_isw)
# print(df_is)
# print(df_ib)
```

In [6]:
```
# Change column type

df_csw['Concentration'] = df_csw['Concentration'].astype(float)
df_cs['Concentration'] = df_cs['Concentration'].astype(float)
df_cb['Concentration'] = df_cb['Concentration'].astype(float)
df_isw['Concentration'] = df_isw['Concentration'].astype(float)
df_is['Concentration'] = df_is['Concentration'].astype(float)
df_ib['Concentration'] = df_ib['Concentration'].astype(float)
```

In [7]:
```
# print(df_csw)
# print(df_cs)
# print(df_cb)
# print(df_isw)
# print(df_is)
# print(df_ib)
```

EDA

In [8]:
```
# Check for Duplicates

print("duplicates")
print("csw :" , df_csw.duplicated().sum())
print("cs  :" , df_cs.duplicated().sum())
print("cb  :" , df_cb.duplicated().sum())
print("isw :" , df_isw.duplicated().sum())
print("is  :" , df_is.duplicated().sum())
print("ib  :" , df_ib.duplicated().sum())
```

```
duplicates
csw : 26
cs  : 1
cb  : 47
isw : 7
is  : 0
ib  : 1
```

In [9]:
```
# Removing Duplicates

df_csw.drop_duplicates(inplace=True)
df_cs.drop_duplicates(inplace=True)
df_cb.drop_duplicates(inplace=True)
df_isw.drop_duplicates(inplace=True)
df_ib.drop_duplicates(inplace=True)
```

In [10]:
```
# print("duplicates")
# print("csw :" , df_csw.duplicated().sum())
# print("cs  :" , df_cs.duplicated().sum())
# print("cb  :" , df_cb.duplicated().sum())
# print("isw :" , df_isw.duplicated().sum())
# print("is  :" , df_is.duplicated().sum())
# print("ib  :" , df_ib.duplicated().sum())
```

In [11]:
```
# Check for null values

print("NAs")
print("csw")
print(df_csw.isnull().sum())
print("cs")
print(df_cs.isnull().sum())
print("cb")
print(df_cb.isnull().sum())
print("isw")
print(df_isw.isnull().sum())
print("is")
print(df_is.isnull().sum())
print("ib")
print(df_ib.isnull().sum())
```

```
NAs
csw
Region           9
Contaminants     0
Concentration    0
dtype: int64
cs
Region           0
Contaminants     0
Concentration    0
dtype: int64
cb
Region           0
Contaminants     0
Concentration    0
dtype: int64
isw
Region           0
Contaminants     0
Concentration    0
dtype: int64
is
Region           0
Contaminants     0
Concentration    0
dtype: int64
ib
Region           0
Contaminants     0
Concentration    0
dtype: int64
```

In [12]:
```
# Removing NAs

df_csw.dropna(inplace=True)
df_cs.dropna(inplace=True)
df_cb.dropna(inplace=True)
df_isw.dropna(inplace=True)
df_ib.dropna(inplace=True)
```

In [13]:
```
# print("NAs")
# print("csw")
# print(df_csw.isnull().sum())
# print("cs")
# print(df_cs.isnull().sum())
# print("cb")
# print(df_cb.isnull().sum())
# print("isw")
# print(df_isw.isnull().sum())
# print("is")
# print(df_is.isnull().sum())
# print("ib")
# print(df_ib.isnull().sum())
```

In [14]:
```
# Check for Outliers

z = np.abs(stats.zscore(df_csw['Concentration']))
threshold = 3
print("df_csw :",np.where(z > 3))

z = np.abs(stats.zscore(df_cs['Concentration']))
threshold = 3
print("df_cs :",np.where(z > 3))

z = np.abs(stats.zscore(df_cb['Concentration']))
threshold = 3
print("df_cb :",np.where(z > 3))

z = np.abs(stats.zscore(df_isw['Concentration']))
threshold = 3
print("df_isw :",np.where(z > 3))

z = np.abs(stats.zscore(df_is['Concentration']))
threshold = 3
print("df_is :",np.where(z > 3))

z = np.abs(stats.zscore(df_ib['Concentration']))
threshold = 3
print("df_ib :",np.where(z > 3))
```

```
df_csw : (array([478, 479, 480, 481, 482, 492, 625]),)
df_cs : (array([ 85, 220, 222]),)
df_cb : (array([ 31,  72, 294, 302, 310, 318, 321, 322]),)
df_isw : (array([100, 101, 178]),)
df_is : (array([7, 8]),)
df_ib : (array([43]),)
```

In [15]:
```
# Removing Outliers

df_csw_filtered = df_csw[df_csw['Concentration'] < 478]
df_cs_filtered = df_cs[df_cs['Concentration'] < 85]
df_cb_filtered = df_cb[df_cb['Concentration'] < 31]
df_isw_filtered = df_isw[df_isw['Concentration'] < 100]
df_is_filtered = df_is[df_is['Concentration'] < 7]
df_ib_filtered = df_ib[df_ib['Concentration'] < 43]
```

Checking Normality

First we have to check wether the data set follows a normal distribution or not. To that we can use Shapiro-Wilk test.

Hypothesis of Shapiro-Wilk test are

Shapiro-Wilk Test

- H_0 : The population from which the sample is drawn follows a normal distribution.
- H_1 : The population from which the sample is drawn does not follow a normal distribution.

In [16]:
```
print("csw :", shapiro(df_csw_filtered["Concentration"]))
print("cs :", shapiro(df_cs_filtered["Concentration"]))
print("cb :", shapiro(df_cb_filtered["Concentration"]))
print("isw :", shapiro(df_isw_filtered["Concentration"]))
print("is :", shapiro(df_is_filtered["Concentration"]))
print("ib :", shapiro(df_ib_filtered["Concentration"]))
```

```
csw : ShapiroResult(statistic=0.25874900817871094, pvalue=1.1070257868166055e-43)
cs : ShapiroResult(statistic=0.4994671940803528, pvalue=1.4910196799080168e-26)
cb : ShapiroResult(statistic=0.5931564569473267, pvalue=2.381692237929861e-30)
isw : ShapiroResult(statistic=0.5712572932243347, pvalue=4.0107722203457513e-25)
is : ShapiroResult(statistic=0.5033602714538574, pvalue=3.216002835673866e-14)
ib : ShapiroResult(statistic=0.6832447052001953, pvalue=6.833248988868945e-08)
```

Since all p-value is less than .05, we reject the null hypothesis of the Shapiro-Wilk test.

** None of above datasets are not in normal distribution.

Statistical Test

So we have to go with non parametric tests. Here the suitable test is Kruskal-Wallis test.

Hypothesis of Kruskal-Wallis test

- H_0 : The independent samples all have the same central tendency and therefore come from the same population.
- H_1 : At least one of the independent samples does not have the same central tendency as the other samples and therefore originates from a different population.

In [17]:
```
print("Region vs Concentration")
print("csw")
print(kruskal(data=df_csw_filtered, dv='Concentration', between='Region'), "\n")
print("cs")
print(kruskal(data=df_cs_filtered, dv='Concentration', between='Region'), "\n")
print("cb")
print(kruskal(data=df_cb_filtered, dv='Concentration', between='Region'), "\n")
print("isw")
print(kruskal(data=df_isw_filtered, dv='Concentration', between='Region'), "\n")
print("is")
print(kruskal(data=df_is_filtered, dv='Concentration', between='Region'), "\n")
print("ib")
print(kruskal(data=df_ib_filtered, dv='Concentration', between='Region'), "\n")
```

```
Region vs Concentration
csw
         Source  ddof1          H         p-unc
Kruskal  Region      4  59.103711  4.475380e-12

cs
         Source  ddof1         H         p-unc
Kruskal  Region      2  43.86028  2.991309e-10

cb
         Source  ddof1          H         p-unc
Kruskal  Region      2  43.731934  3.189563e-10

isw
         Source  ddof1          H     p-unc
Kruskal  Region      3  18.753395  0.000307

is
         Source  ddof1        H     p-unc
Kruskal  Region      1  2.77906  0.095504

ib
         Source  ddof1         H     p-unc
Kruskal  Region      1  1.655172  0.198256
```

In [18]:
```
print("Contaminants vs Concentration")
print("csw")
print(kruskal(data=df_csw_filtered, dv='Concentration', between='Contaminants '), "\n")
print("cs")
print(kruskal(data=df_cs_filtered, dv='Concentration', between='Contaminants '), "\n")
print("cb")
print(kruskal(data=df_cb_filtered, dv='Concentration', between='Contaminants '), "\n")
print("isw")
print(kruskal(data=df_isw_filtered, dv='Concentration', between='Contaminants '), "\n")
print("is")
print(kruskal(data=df_is_filtered, dv='Concentration', between='Contaminants '), "\n")
print("ib")
print(kruskal(data=df_ib_filtered, dv='Concentration', between='Contaminants '), "\n")
```

```
Contaminants vs Concentration
csw
                Source  ddof1           H         p-unc
Kruskal  Contaminants     167  303.568703  5.368023e-10

cs
                Source  ddof1           H         p-unc
Kruskal  Contaminants      93  214.489278  1.358967e-11

cb
                Source  ddof1           H         p-unc
Kruskal  Contaminants      55  343.839608  2.114419e-43

isw
                Source  ddof1           H         p-unc
Kruskal  Contaminants     137  240.142718  1.222952e-07

is
                Source  ddof1          H     p-unc
Kruskal  Contaminants      42  54.077479  0.100206

ib
                Source  ddof1          H     p-unc
Kruskal  Contaminants       7  24.993162  0.000761
```

Test Results

Below tables shows the p value of the test results.

Dataset	Vs Region	Vs Contaminants
csw	4.475380e-12	5.368023e-10
cs	2.991309e-10	1.358967e-11
cb	3.189563e-10	2.114419e-43
isw	0.000307	1.222952e-07
is	0.095504	0.100206
ib	0.198256	0.000761

Result Interpretation

If p value of the Kruskal-Wallis test is below than 0.05, we can say that there is a relationship between parameters which was tested.

With test results we can say that,

1. In ***csw, cs, cb, isw datasets*** there is a relationship between
    - region & Concentration
    - Contaminants & Concentration
      (But we can't say that there is a relationship between region & Contaminants in those dataset.)
2. In ***is dataset*** there is a relationship between region & Concentration.
3. In ***ib dataset*** there is a relationship between Contaminants & Concentration.