Opdracht week 2

#with open(INPUT\_CSV, newline='') as csvfile:

#reader = csv.DictReader(csvfile)

# reader = pandas.read\_csv()

# print(reader)

# Country, Region, Pop. Density, Infant mortality and GDP uit csv\_file halen

#for row in reader:

# country = row[1]

# print(country)

# df = pandas.DataFrame(data,column=['Country', 'Region', 'pop\_density', 'infant\_mortality', 'GDP'])

#for line in lines:

#striped\_lines = line.rstrip()

# space =

# find inconsistencies + clean rows with many missing columns + clean rows with missing values

#space = line[i]

# def check\_empty(list):

# if lines == '':

# print(x)

# Country”, “Region”, “Pop. Density (per sq. mi.)”, “Infant mortality (per 1000 births)” and “GDP ($ per capita) dollars

# analyze Central Tendency and the Five Number Summary

# Central Tendency: the mean, median and mode.

Mask/t – maakt waardes in een NUNN

Outliers = 3\*standaard deviatie + mean

import pandas

#import pandas.io.data as web

import csv

#from requests import get

#from requests.exceptions import RequestException

#from contextlib import closing

from bs4 import BeautifulSoup

import json

import matplotlib.pyplot as plt

INPUT\_CSV = 'input.csv'

# load data into pandas DataFrame

reader = pandas.read\_csv(INPUT\_CSV)

#print(reader)

# preprocess all usable rows + dollars eraf halen + string omzetten naar nummer (int/float)

reader['GDP ($ per capita) dollars'] = reader['GDP ($ per capita) dollars'].str.strip(' dollars')

reader['GDP ($ per capita) dollars'] = pandas.to\_numeric(reader['GDP ($ per capita) dollars'], errors='coerce')

#reader['GDP ($ per capita) dollars'] = reader['GDP ($ per capita) dollars'].apply(lambda x: x.astype('float64', errors='ignore'))

print(reader['GDP ($ per capita) dollars'])

# calculate the standard deviation, mean, median and mode of the GDP

mean = reader['GDP ($ per capita) dollars'].mean(axis=0)

print(mean)

standard\_deviation = reader['GDP ($ per capita) dollars'].std()

print(standard\_deviation)

median = reader['GDP ($ per capita) dollars'].median()

print(median)

mode = reader['GDP ($ per capita) dollars'].mode()

print(mode)

# bereken outliers

outliers\_max = mean+3\*standard\_deviation

outliers\_min = mean-3\*standard\_deviation

#print(outliers)

reader['GDP ($ per capita) dollars'].mask(reader['GDP ($ per capita) dollars'] > outliers\_max, inplace = True)

reader['GDP ($ per capita) dollars'].mask(reader['GDP ($ per capita) dollars'] < outliers\_min, inplace = True)

# plot a histogram

reader['GDP ($ per capita) dollars'].hist()

plt.show()

# Five number summary:calculate Minimum, First Quartile, Median, Third Quartile and Maximum of Infant Mortality

reader['Infant mortality (per 1000 births)'] = pandas.to\_numeric(reader['Infant mortality (per 1000 births)'], errors='coerce')

minimum = reader['Infant mortality (per 1000 births)'].min()

print(minimum)

first\_quartile = reader['Infant mortality (per 1000 births)'].quantile(q=0.25)

print(first\_quartile)

median = reader['Infant mortality (per 1000 births)'].median()

print(median)

third\_quartile = reader['Infant mortality (per 1000 births)'].quantile(q=0.75)

print(third\_quartile)

maximum = reader['Infant mortality (per 1000 births)'].max()

print(maximum)

# create a box plot

reader['GDP ($ per capita) dollars'].boxplot()

plt.show()

# cleaned, preprocessed and analyzed data to a .json

# if \_\_name\_\_ == "\_\_main\_\_":

#print(reader)

#print(data\_dict)

#df['lines'].plot(0)

#plt.show()

Alle dingen die ik heb verwijderd:

#from requests import get

#from requests.exceptions import RequestException

#from contextlib import closing

#import pandas.io.data as web

#print(reader['GDP ($ per capita) dollars'])

#reader['GDP ($ per capita) dollars'] = reader['GDP ($ per capita) dollars'].apply(lambda x: x.astype('float64', errors='ignore'))

#print(outliers)

# minimum = reader['Infant mortality (per 1000 births)'].min()

# print(minimum)

# first\_quartile = reader['Infant mortality (per 1000 births)'].quantile(q=0.25)

# print(first\_quartile)

# median = reader['Infant mortality (per 1000 births)'].median()

# print(median)

# third\_quartile = reader['Infant mortality (per 1000 births)'].quantile(q=0.75)

# print(third\_quartile)

# maximum = reader['Infant mortality (per 1000 births)'].max()

# print(maximum)

“Country”, “Region”, “Pop. Density (per sq. mi.)”, “Infant mortality (per 1000 births)” and “GDP ($ per capita) dollars”.

Population,Area (sq. mi.), Coastline (coast/area ratio),Net migration, Literacy (%),Phones (per 1000),Arable (%),Crops (%),Other (%),Climate,Birthrate,Deathrate,Agriculture,Industry,Service

Drop.

import pandas

import csv

from bs4 import BeautifulSoup

import json

import matplotlib.pyplot as plt

# load data into pandas DataFrame

INPUT\_CSV = 'input.csv'

reader = pandas.read\_csv(INPUT\_CSV)

def GDP(reader):

# preprocess all usable rows + dollars eraf halen + string omzetten naar nummer (int/float)

reader['GDP ($ per capita) dollars'] = reader['GDP ($ per capita) dollars'].str.strip(' dollars')

reader['GDP ($ per capita) dollars'] = pandas.to\_numeric(reader['GDP ($ per capita) dollars'], errors='coerce')

# calculate the standard deviation, mean, median and mode of the GDP

mean = reader['GDP ($ per capita) dollars'].mean(axis=0)

print(mean)

standard\_deviation = reader['GDP ($ per capita) dollars'].std()

print(standard\_deviation)

median = reader['GDP ($ per capita) dollars'].median()

print(median)

mode = reader['GDP ($ per capita) dollars'].mode()

print(mode)

# bereken outliers

outliers\_max = mean+3\*standard\_deviation

outliers\_min = mean-3\*standard\_deviation

reader['GDP ($ per capita) dollars'].mask(reader['GDP ($ per capita) dollars'] > outliers\_max, inplace = True)

reader['GDP ($ per capita) dollars'].mask(reader['GDP ($ per capita) dollars'] < outliers\_min, inplace = True)

# plot a histogram

def histogram(reader):

reader['GDP ($ per capita) dollars'].hist(bins=50)

plt.xlabel('GDP', fontsize=12, fontweight='bold')

plt.ylabel('Frequentie', fontsize=12, fontweight='bold')

plt.title('by Liora Rosenberg')

plt.suptitle('GDP ($ per capita) dollars', fontsize=16, fontweight='bold')

plt.show()

def mortality(reader):

# Five number summary:calculate Minimum, First Quartile, Median, Third Quartile and Maximum of Infant Mortality

reader['Infant mortality (per 1000 births)'] = reader['Infant mortality (per 1000 births)'].str.replace(",", ".")

reader['Infant mortality (per 1000 births)'] = pandas.to\_numeric(reader['Infant mortality (per 1000 births)'], errors='coerce')

print(reader['Infant mortality (per 1000 births)'])

five\_number\_summary = reader['Infant mortality (per 1000 births)'].describe()

print(five\_number\_summary)

# create a box plot

print(type(reader['Infant mortality (per 1000 births)']))

plt.style.use('ggplot')

reader['Infant mortality (per 1000 births)'].plot.box()

plt.title('by Liora Rosenberg')

plt.suptitle('Infant mortality (per 1000 births)', fontsize=16, fontweight='bold')

#plt.axis(0,20)

plt.show()

def jason(reader):

# cleaned, preprocessed and analyzed data to a .json

#pandas.reader.to\_json()

reader.set\_index("Country")

file = reader.to\_json('x.json', orient = "index")

print(file)

# alleen rechterkant stripen

if \_\_name\_\_ == "\_\_main\_\_":

GDP(reader)

histogram(reader)

mortality(reader)

jason(reader)

#pandas.reader.to\_json()

#reader.set\_index("Country")

Frame

Replace(“,” “.”)

Frame\_set index country

#from bs4 import BeautifulSoup

Voeg de wortel van de graaf toe aan de queue

search root of the graph and add to queue

if there is a node:

if there is a solution:

stop searching and give a solution

elif there is not a solution:

add all children

if queue is empty

stop searching

Als er een knoop in de queue zit, neem deze uit de queue en bekijk de knoop:

Als dit een oplossing is: stop het zoeken en geef de oplossing

Als dit geen oplossing is: voeg alle kinderen van deze knoop toe aan het einde van de FIFO queue

Als de queue leeg is: alle knopen zijn bekeken dus stop het zoeken en geef aan dat er geen oplossing is

Ga door naar stap 2

print(reader['Infant mortality (per 1000 births)'])