

Python script 1 to generate random systems

```
# -*- coding: utf-8 -*-

import csv, random

Health_costs_list = [1,2,3]
Total_nitrogen_emissions_list = [1,2,3]
Suspended_solid_emissions_list = [1,2,3]
On_farm_ground_surface_used_list = [1,2,3]
Global_warming_potential_list = [1,2,3,4,5]
Acidification_potential_list = [1,2,3]
Eutrophication_potential_list = [1,2,3]
Fish_in_fish_out_ratio_list = [1,2,3,4,5]
Percentage_of_wild_juveniles_and_plant_used_list = [1,2,3]
Percentage_of_renewability_list = [1,2,3]
Water_demand_list = [1,2,3,4]
Net_primary_production_use_list = [1,2,3]
Global_land_competition_list = [1,2,3,4,5]
Total_cumulative_energy_demand_list = [1,2,3,4,5]
Percentage_of_nitrogen_derived_from_coproducts_list = [1,2,3]
Percentage_of_phosphorus_recovered_list = [1,2,3]
Production_loss_list = [1,2,3,4,5]
On_farm_energy_efficiency_list = [1,2,3,4,5]
Percentage_of_renewable_energy_used_list = [1,2,3]
Total_feed_conversion_rate_list = [1,2,3,4,5]
Nitrogen_use_efficiency_list = [1,2,3]
Feedstuff_locally_produced_list = [1,2,3]
Emergy_Yield_ratio_list = [1,2,3]
Predator_control_list = [1,2]
Biosecurity_and_good_practices_list = [1,2,3]
Multitrophic_integration_list = [1,2,3]
Production_diversification_list = [1,2,3]
Escapees_management_list = [1,2,3]
Interactions_with_professional_institutions_list = [1,2,3]
Professional_involvement_list = [1,2,3]
Independance_towards_suppliers_list = [1,2,3]
Independance_towards_customers_list = [1,2,3]
Workload_list = [1,2,3]
Health_and_safety_list = [1,2,3]
Job_difficulty_appreciation_list = [1,2,3]
Labour_remuneration_list = [1,2,3]
Working_status_list = [1,2,3]
Education_level_list = [1,2,3]
Gender_equality_list = [1,2,3]
Employment_of_workers_with_handicap_list = [1,2]
Fish_physical_damages_list = [1,2,3]
Stocking_density_list = [1,2,3]
Assured_supply_of_food_products_list = [1,2,3,4,5]
Accessibility_of_products_list = [1,2,3,4,5]
Nutritional_quality_list = [1,2,3,4,5]
Contribution_to_employment_list = [1,2,3,4,5]
Education_contribution_list = [1,2,3]
Labour_productivity_list = [1,2,3,4,5]
Average_sales_prices_list = [1,2,3,4,5]
Paid_labour_costs_list = [1,2,3,4,5]
Feed_costs_list = [1,2,3,4,5]
Juvenile_and_seedling_costs_list = [1,2,3,4,5]
Net_present_value_list = [1,2,3]
Internal_rate_of_return_list = [1,2,3]
Subsidies_weight_list = [1,2,3]
Resistance_to_environmental_constraints_list = [1,2,3,4,5]
Specialization_rate_list = [1,2,3]
```

```
Option = []
Option.append(Health_costs_list)
Option.append(Total_nitrogen_emissions_list)
Option.append(Suspended_solid_emissions_list)
Option.append(On_farm_ground_surface_used_list)
Option.append(Global_warming_potential_list)
Option.append(Acidification_potential_list)
Option.append(Eutrophication_potential_list)
Option.append(Fish_in_fish_out_ratio_list)
Option.append(Percentage_of_wild_juveniles_and_plant_used_list)
Option.append(Percentage_of_renewability_list)
Option.append(Water_demand_list)
Option.append(Net_primary_production_use_list)
Option.append(Global_land_competition_list)
Option.append(Total_cumulative_energy_demand_list)
Option.append(Percentage_of_nitrogen_derived_from_coproducts_list)
Option.append(Percentage_of_phosphorus_recovered_list)
Option.append(Production_loss_list)
Option.append(On_farm_energy_efficiency_list)
Option.append(Percentage_of_renewable_energy_used_list)
Option.append(Total_feed_conversion_rate_list)
Option.append(Nitrogen_use_efficiency_list)
Option.append(Feedstuff_locally_produced_list)
Option.append(Emergy_Yield_ratio_list)
Option.append(Predator_control_list)
Option.append(Biosecurity_and_good_practices_list)
Option.append(Multitrophic_integration_list)
Option.append(Production_diversification_list)
Option.append(Escapees_management_list)
Option.append(Interactions_with_professional_institutions_list)
Option.append(Professional_involvement_list)
Option.append(Independance_towards_suppliers_list)
Option.append(Independance_towards_customers_list)
Option.append(Workload_list)
Option.append(Health_and_safety_list)
Option.append(Job_difficulty_appreciation_list)
Option.append(Labour_remuneration_list)
Option.append(Working_status_list)
Option.append(Education_level_list)
Option.append(Gender_equality_list)
Option.append(Employment_of_workers_with_handicap_list)
Option.append(Production_loss_list)
Option.append(Fish_physical_damages_list)
Option.append(Stocking_density_list)
Option.append(Biosecurity_and_good_practices_list)
Option.append(Assured_supply_of_food_products_list)
Option.append(Accessibility_of_products_list)
Option.append(Nutritional_quality_list)
Option.append(Fish_physical_damages_list)
Option.append(Contribution_to_employment_list)
Option.append(Feedstuff_locally_produced_list)
Option.append(Education_contribution_list)
Option.append(On_farm_energy_efficiency_list)
Option.append(Labour_productivity_list)
Option.append(Total_feed_conversion_rate_list)
Option.append(Production_loss_list)
Option.append(Nutritional_quality_list)
Option.append(Fish_physical_damages_list)
Option.append(Average_sales_prices_list)
Option.append(Paid_labour_costs_list)
Option.append(Feed_costs_list)
Option.append(Juvenile_and_seedling_costs_list)
Option.append(Net_present_value_list)
Option.append(Internal_rate_of_return_list)
Option.append(Subsidies_weight_list)
Option.append(Emergy_Yield_ratio_list)
Option.append(Production_diversification_list)
Option.append(Biosecurity_and_good_practices_list)
Option.append(Resistance_to_environmental_constraints_list)
Option.append(Specialization_rate_list)
Option.append(Independance_towards_suppliers_list)
Option.append(Independance_towards_customers_list)
Option.append(Fish_in_fish_out_ratio_list)
```

```
iterations = 10000

def getElem(file, i, j):
    with open(file, 'r') as f:
        reader = csv.reader(f)
        for line in reader:
            if reader.line_num - 1 == i:
                return line[j]

print(getElem('BDV.csv',0,0))

list_export = []
first_row = [""]
for i in range(0,iterations):
    first_row.append('System '+str(i+1))
    list_export.append(first_row)

for i in range(1,len(Option)+1):
    list_prov = []
    list_prov.append(getElem('BDV.csv',i,0))
    for j in range(0,iterations):
        list_prov.append(str(random.choice(Option[i-1])))
        j=j+1
    list_export.append(list_prov)
    i=i+1

print(list_export)

with open('export_monte_carlo.csv', 'w', newline='') as f:
    writer = csv.writer(f,delimiter=',', quotechar='"', quoting=csv.QUOTE_ALL)
    writer.writerows(list_export)
```

Python script 2 to export DEXiAqua results to Excel

```
import csv

import xlrd

from xlwt import Workbook

def getElem(file, i, j):
    with open(file, 'r') as f:
        reader = csv.reader(f)
        for line in reader:
            if reader.line_num - 1 == i:
                return line[j]

iterations = 10000
resultats = []

for i in range(1, iterations+1):
    resultats.append(getElem('resBDV.csv', 1, i))
    i=i+1

path = r"resultats.xls"
# On créer un "classeur"
classeur = Workbook()
# On ajoute une feuille au classeur
feuille = classeur.add_sheet("répartition")

for i in range(0, len(resultats)):
    feuille.write(i, 0, resultats[i])

# Ecriture du classeur sur le disque
classeur.save(path)
```