DAT565/DIT407 Assignment 1

Ola Bratt
ola.bratt@gmail.com

2024-01-16

Problem 1: Dependency Ratio

Results are presented in Figure 1 and Figure 2.

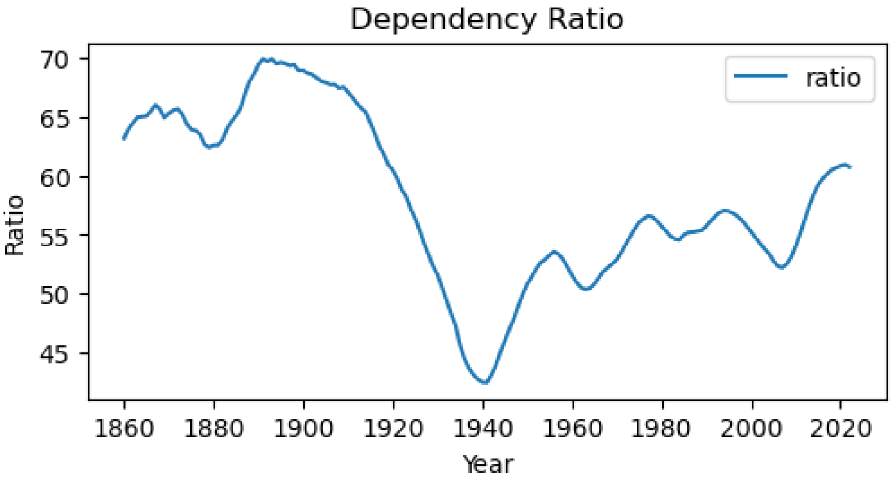

Figure 1: Dependecy ratio

If you need to cite external sources, do so by placing the literature assignment, we use data from SCB [1].

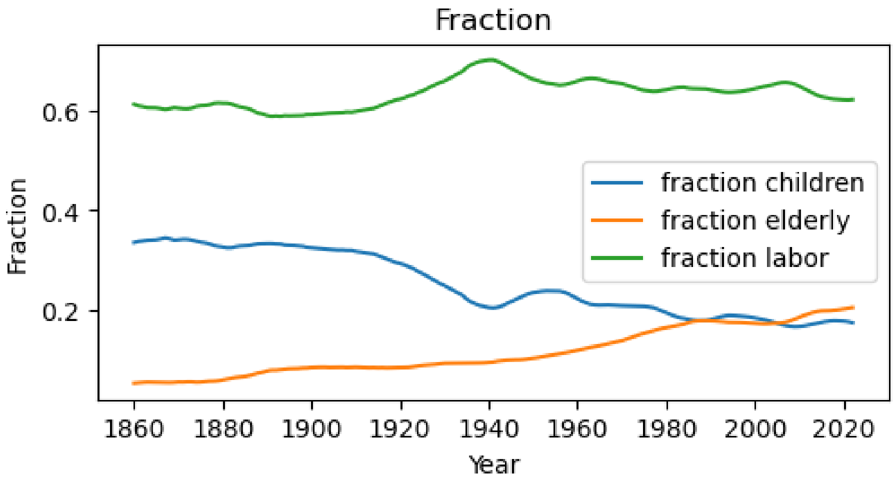

Figure 2: Fractions

Discussion of results.

References

[1] Statistiska centralbyrån. *Folkmängden efter ålder och kön. År 1860 - 2022.* Retrieved 2023-10-20. 2023. URL: https://www.statistikdatabasen.scb.se/pxweb/sv/ssd/START__BE__BE0101__BE0101A/BefolkningR1860N/.

Appendix: Source Code

```
{
 ”cells”: [
  {
   ”cell_type”: ”markdown”,
   ”id”: ”7030bacd”,
   ”metadata”: {},
   ”source”: [
    ”<h1>Import frameworks</h1>\n”,
    ”<p>numpy – numerical calculations</p>\n”,
    ”<p>pandas – data analys</p>\n”,
    ”<p>matplotlib – graphs and plotting</p>”
   ]
  },
  {
   ”cell_type”: ”code”,
   ”execution_count”: 49,
   ”id”: ”27ff796d”,
   ”metadata”: {},
   ”outputs”: [],
   ”source”: [
    ”import numpy as np\n”,
    ”import pandas as pd\n”,
    ”from matplotlib import pyplot”
   ]
  },
  {
   ”cell_type”: ”markdown”,
   ”id”: ”018dea49”,
   ”metadata”: {},
   ”source”: [
    ”<h1>Lamda functions</h1>”
   ]
  },
  {
   ”cell_type”: ”code”,
   ”execution_count”: 50,
   ”id”: ”1162d384”,
   ”metadata”: {},
   ”outputs”: [],
   ”source”: [
    ”ratio = lambda children, elderly, labor: 100 * (children + elderly) / labo
    ”total = lambda children, elderly, labor: children + elderly + labor\n”,
    ”fraction = lambda part, total: part / total”
   ]
  },
  {
   ”cell_type”: ”markdown”,
   ”id”: ”73b62892”,
   ”metadata”: {},
   ”source”: [
    ”<h1>Load population dataFrame</h1>”
   ]
  },
  {
   ”cell_type”: ”code”,
   ”execution_count”: 51,
   ”id”: ”2684af35”,
   ”metadata”: {},
   ”outputs”: [],
   ”source”: [
    ”population = pd.read_csv('swedish_population_by_year_and_sex_1860-2022.csv
sep=',',)\n”,
    ”# Drop sex column, we don't need it\n”,
    ”populationNoSex = population.drop(columns=['sex'])\n”,
    ”\n”,
   ”# Set age to numeric\n”,
   ”populationNoSex.at[220,'age'] = 110\n”,
   ”populationNoSex.at[221,'age'] = 110\n”,
   ”# Convert to numeric\n”,
   ”populationNoSex['age'] = pd.to_numeric(populationNoSex['age'], errors='coe
   ”\n”,
   ”\n”,
   ”# Group by age\n”,
   ”classes = populationNoSex.groupby(pd.cut(populationNoSex['age'], [−1, 14,
   ”# Drop age column, we don't need it anymore\n”,
   ”classes = classes.drop(columns=['age'])\n”,
   ”# Transpose\n”,
   ”classesT = classes.transpose()\n”,
   ”# Apply lambda functions\n”,
   ”classesT['ratio'] = classesT.apply(lambda row: ratio(row.iat[0], row.iat[2
   ”classesT['total'] = classesT.apply(lambda row: total(row.iat[0], row.iat[2
   ”classesT['fraction_children'] = classesT.apply(lambda row: fraction(row.ia
   ”classesT['fraction_elderly'] = classesT.apply(lambda row: fraction(row.iat
   ”classesT['fraction_labor'] = classesT.apply(lambda row: fraction(row.iat[1
   ”\n”,
   ”#print(classesT.head())\n”,
   ”#print(classesT.tail())\n”,
   ”\n”,
   ”\n”
  ]
 },
 {
  ”cell_type”: ”markdown”,
  ”id”: ”278c2add”,
  ”metadata”: {},
  ”source”: [
   ”<h1>Plot</h1>”
  ]
 },
 {
  ”cell_type”: ”code”,
  ”execution_count”: 52,
  ”id”: ”34367ae7”,
  ”metadata”: {},
  ”outputs”: [
   {
    ”data”: {
     ”text/plain”: [
      ”<matplotlib.legend.Legend at 0x7f7b8ab65be0>”
     ]
    },
    ”execution_count”: 52,
    ”metadata”: {},
    ”output_type”: ”execute_result”
   },
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAAf8AAAEZCAYAAACZ7CwhAAAAOXRFWH
      "text/plain": [
       "<Figure size 500x270 with 1 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    },
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAAf8AAAEZCAYAAACZ7CwhAAAAOXRFWH
      "text/plain": [
       "<Figure size 500x270 with 1 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    }
   ],
   "source": [
    "# Convert index to float\n",
    "years = np.asarray(classesT.index.values, float)\n",
    "\n",
    "# Plot ratio\n",
    "fig1, ax1 = pyplot.subplots(figsize=(5, 2.7), layout='constrained')\n",
    "ax1.plot(years, classesT['ratio'], label='ratio')\n",
    "ax1.set_xlabel('Year')  # Add an x-label to the axes.\n",
    "ax1.set_ylabel('Ratio')  # Add a y-label to the axes.\n",
    "ax1.set_title(\"Dependency Ratio\")  # Add a title to the axes.\n",
    "ax1.legend()\n",
    "\n",
    "# Plot fractions\n",
    "fig2, ax2 = pyplot.subplots(figsize=(5, 2.7), layout='constrained')\n",
    "ax2.plot(years, classesT['fraction_children'], label='fraction children')\
    "ax2.plot(years, classesT['fraction_elderly'], label='fraction elderly')\n"
    "ax2.plot(years, classesT['fraction_labor'], label='fraction labor')\n",
    "ax2.set_xlabel('Year')  # Add an x-label to the axes.\n",
    "ax2.set_ylabel('Fraction')  # Add a y-label to the axes.\n",
    "ax2.set_title(\"Fraction of total population\")  # Add a title to the axes
    "ax2.legend()\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "19f0847c",
   "metadata": {},
   "source": []
  }
```

```
 ],
 ”metadata”: {
  ”interpreter”: {
   ”hash”: ”ced63c54952fb711462c5b23e25925157bd2f9b85cfe19b36cd014c2f46d5e7f”
  },
  ”kernelspec”: {
   ”display_name”: ”Python 3.8.8 64−bit (’base’: conda)”,
   ”name”: ”python3”
  },
  ”language_info”: {
   ”codemirror_mode”: {
    ”name”: ”ipython”,
    ”version”: 3
   },
   ”file_extension”: ”.py”,
   ”mimetype”: ”text/x−python”,
   ”name”: ”python”,
   ”nbconvert_exporter”: ”python”,
   ”pygments_lexer”: ”ipython3”,
   ”version”: ”3.8.10”
  }
 },
 ”nbformat”: 4,
 ”nbformat_minor”: 5
}
```