2023-05-12_BasicDataTransformations

May 15, 2023

[18]:

```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.cm as cm

%matplotlib inline
colors=plt.rcParams['axes.prop_cycle'].by_key()['color']
```

1 Basic transformations on tabular data

As example data use UN World Population Example Dataset (see other files)

1.1 Import data

[19]:

```
# set up column data types
headerLine="SortOrder,LocID,Notes,ISO3_code,ISO2_code,SDMX_code,LocTypeID,LocTypeName,"+\
    "ParentID,Location,VarID,Variant,"+\
  ␣
 ↪"Time,TPopulation1Jan,TPopulation1July,TPopulationMale1July,TPopulationFemale1July,"+\
  ␣
 ↪"PopDensity,PopSexRatio,MedianAgePop,NatChange,NatChangeRT,PopChange,PopGrowthRate,"+\
  ␣
 ↪"DoublingTime,Births,Births1519,CBR,TFR,NRR,MAC,SRB,Deaths,DeathsMale,DeathsFemale,"+\
  ␣
 ↪"CDR,LEx,LExMale,LExFemale,LE15,LE15Male,LE15Female,LE65,LE65Male,LE65Female,"+\
    "LE80,LE80Male,LE80Female,InfantDeaths,IMR,LBsurvivingAge1,Under5Deaths,"+\
  ␣
 ↪"Q5,Q0040,Q0040Male,Q0040Female,Q0060,Q0060Male,Q0060Female,Q1550,Q1550Male,"+\
    "Q1550Female,Q1560,Q1560Male,Q1560Female,NetMigrations,CNMR"

dtypeDict={}
for x in headerLine.split(","):
    dtypeDict[x]=np.float64
```

```
for x in "SortOrder,LocID,LocTypeID,ParentID,VarID,Time".split(","):
    dtypeDict[x]=np.int32
for x in "Notes,ISO3_code,SDMX_code,ISO2_code,LocTypeName,Location,Variant".
 ↪split(","):
    dtypeDict[x]=str
```

[20]:
```
dataFull=pd.read_csv("data/WPP2022_Demographic_Indicators_Medium.
 ↪csv",sep=",",dtype=dtypeDict)
```

[21]:
```
# check list of columns
dataFull.keys()
```

[21]:
```
Index(['SortOrder', 'LocID', 'Notes', 'ISO3_code', 'ISO2_code', 'SDMX_code',
       'LocTypeID', 'LocTypeName', 'ParentID', 'Location', 'VarID', 'Variant',
       'Time', 'TPopulation1Jan', 'TPopulation1July', 'TPopulationMale1July',
       'TPopulationFemale1July', 'PopDensity', 'PopSexRatio', 'MedianAgePop',
       'NatChange', 'NatChangeRT', 'PopChange', 'PopGrowthRate',
       'DoublingTime', 'Births', 'Births1519', 'CBR', 'TFR', 'NRR', 'MAC',
       'SRB', 'Deaths', 'DeathsMale', 'DeathsFemale', 'CDR', 'LEx', 'LExMale',
       'LExFemale', 'LE15', 'LE15Male', 'LE15Female', 'LE65', 'LE65Male',
       'LE65Female', 'LE80', 'LE80Male', 'LE80Female', 'InfantDeaths', 'IMR',
       'LBsurvivingAge1', 'Under5Deaths', 'Q5', 'Q0040', 'Q0040Male',
       'Q0040Female', 'Q0060', 'Q0060Male', 'Q0060Female', 'Q1550',
       'Q1550Male', 'Q1550Female', 'Q1560', 'Q1560Male', 'Q1560Female',
       'NetMigrations', 'CNMR'],
      dtype='object')
```

1.2 Transformations

1.2.1 Filtering rows

[22]:
```
# keep only rows for locations that are countries
print(f"rows full data: {dataFull.shape[0]}")

# creating boolean indicators according which to filter
countryIndicator=(dataFull["LocTypeName"]=="Country/Area")
timeIndicator=(dataFull["Time"]<=2022)

# apply logical operation and apply filter (apply copy to avoid later issues)
data=dataFull[countryIndicator & timeIndicator]

# apply copy to avoid later issues (will be illustrated below)
# clear original full dataset from memory
data=data.copy()
del dataFull
```

```
print(f"remaining columns: {data.shape[0]}")
```

```
rows full data: 43472
remaining columns: 17301
```

1.2.2 Select colums

[23]:
```
dataReduced=data[["CBR","CDR"]]
dataReduced.columns
```

[23]:
```
Index(['CBR', 'CDR'], dtype='object')
```

1.2.3 Select rows and columns

(See pandas documentation for more details about accessors.)

[24]:
```
dataSel=data.loc[data["Location"]=="Germany",["Time","CBR"]]
dataSel
```

[24]:
```
       Time     CBR
29184  1950  16.222
29185  1951  15.916
29186  1952  15.733
29187  1953  15.643
29188  1954  15.658
…      …     …
29252  2018   9.423
29253  2019   9.304
29254  2020   9.144
29255  2021   9.167
29256  2022   9.139

[73 rows x 2 columns]
```

1.2.4 Careful when writing on selections of dataframe

[25]:
```
# try this first:
# select rows that are to be changed
dataSel=data[data["Location"]=="United States of America"]
# try to write on them
dataSel["Location"]="USA"
```

```
/tmp/ipykernel_6449/3824782913.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  dataSel["Location"]="USA"
```

```
[26]: # but in the original dataframe the entries have not been changed
      data[data["Location"]=="United States of America"].shape
```

```
[26]: (73, 67)
```

```
[27]: # preferred way to handle this:
      data.loc[data["Location"]=="United States of America","Location"]="USA"
```

```
[28]: # check again
      data[data["Location"]=="United States of America"].shape
```

```
[28]: (0, 67)
```

1.2.5 Generate derived columns

```
[29]: # compute "crude change rate": crude birth rate - crude death rate
      data["CCR"]=data["CBR"]-data["CDR"]
```

1.2.6 Join tables

```
[30]: # have a separate table available, that lists for each country the␣
        ↪corresponding continent
      # now want to add this info as new column into the data frame
      # i.e. for each row in main table, look up row with corresponding country and␣
        ↪copy the given value of continent
      # to a new continent column
      dataContinents=pd.read_csv("data/UN_population/continent_association.csv",\
              dtype={"SDMX_code":str,"Continent":str})
      dataContinents.dtypes
```

```
[30]: SDMX_code    object
      Continent    object
      dtype: object
```

```
[31]: dataContinents["Continent"].unique()
```

```
[31]: array(['Africa', 'Asia', 'Europe', 'Central and South America',
             'North America', 'Oceania'], dtype=object)
```

[32]:

```
# merge with continent data
data=pd.merge(data,dataContinents,on="SDMX_code")
```

[33]:

```
data.shape
```

[33]:

```
(17301, 69)
```

1.2.7 Summaries / reductions

Reduce big dataset to smaller one by computing basic aggregations, summaries, statistics. Typical examples are: * Maximum, minimum, mean, median, variance, … * Count (distinguish: size, count, nunique in pandas)

[35]:

```
[data["Time"].min(),data["Time"].max()]
```

[35]:

```
[1950, 2022]
```

[37]:

```
continents=data["Continent"].unique()
print(continents)
```

```
['Africa' 'Asia' 'Europe' 'Central and South America' 'North America'
 'Oceania']
```

[38]:

```
data["Continent"].nunique()
```

[38]:

```
6
```

[40]:

```
data["TPopulation1Jan"].median()
```

[40]:

```
3121.132
```

[41]:

```
data["TPopulation1Jan"].sum()
```

[41]:

```
365821161.818
```

1.2.8 Grouping

Of course: applying such simplistic summaries to the whole dataset is usually too reductive. Often want to apply them only to subsets separately. This is most conveniently achieved by grouping a dataframe.

[42]:

```
# this splits a dataframe into many smaller, according to distinct values of the given column
dataGrouped=data.groupby("Time")
print("year\t# rows")
for year,datasub in dataGrouped:
    print(f"{year}\t{datasub.shape[0]}")
```

```
year    # rows
1950    237
1951    237
1952    237
1953    237
1954    237
1955    237
1956    237
1957    237
1958    237
1959    237
1960    237
1961    237
1962    237
1963    237
1964    237
1965    237
1966    237
1967    237
1968    237
1969    237
1970    237
1971    237
1972    237
1973    237
1974    237
1975    237
1976    237
1977    237
1978    237
1979    237
1980    237
1981    237
1982    237
1983    237
1984    237
1985    237
1986    237
1987    237
1988    237
1989    237
1990    237
1991    237
1992    237
1993    237
```

```
1994    237
1995    237
1996    237
1997    237
1998    237
1999    237
2000    237
2001    237
2002    237
2003    237
2004    237
2005    237
2006    237
2007    237
2008    237
2009    237
2010    237
2011    237
2012    237
2013    237
2014    237
2015    237
2016    237
2017    237
2018    237
2019    237
2020    237
2021    237
2022    237
```

[43]:
```python
# can group over several columns
# this splits a dataframe into many smaller, according to distinct values of the given column
dataGrouped=data.groupby(["Time","Continent"])
print("year\tcont\t# rows")
for (year,continent),datasub in dataGrouped:
    print(f"{year}\t{continent[:4]}\t{datasub.shape[0]}")
```

```
year    cont    # rows
1950    Afri    58
1950    Asia    51
1950    Cent    50
1950    Euro    50
1950    Nort    5
1950    Ocea    23
1951    Afri    58
1951    Asia    51
1951    Cent    50
```

```
1951    Euro    50
1951    Nort    5
1951    Ocea    23
1952    Afri    58
1952    Asia    51
1952    Cent    50
1952    Euro    50
1952    Nort    5
1952    Ocea    23
1953    Afri    58
1953    Asia    51
1953    Cent    50
1953    Euro    50
1953    Nort    5
1953    Ocea    23
1954    Afri    58
1954    Asia    51
1954    Cent    50
1954    Euro    50
1954    Nort    5
1954    Ocea    23
1955    Afri    58
1955    Asia    51
1955    Cent    50
1955    Euro    50
1955    Nort    5
1955    Ocea    23
1956    Afri    58
1956    Asia    51
1956    Cent    50
1956    Euro    50
1956    Nort    5
1956    Ocea    23
1957    Afri    58
1957    Asia    51
1957    Cent    50
1957    Euro    50
1957    Nort    5
1957    Ocea    23
1958    Afri    58
1958    Asia    51
1958    Cent    50
1958    Euro    50
1958    Nort    5
1958    Ocea    23
1959    Afri    58
1959    Asia    51
1959    Cent    50
```

```
1959    Euro    50
1959    Nort    5
1959    Ocea    23
1960    Afri    58
1960    Asia    51
1960    Cent    50
1960    Euro    50
1960    Nort    5
1960    Ocea    23
1961    Afri    58
1961    Asia    51
1961    Cent    50
1961    Euro    50
1961    Nort    5
1961    Ocea    23
1962    Afri    58
1962    Asia    51
1962    Cent    50
1962    Euro    50
1962    Nort    5
1962    Ocea    23
1963    Afri    58
1963    Asia    51
1963    Cent    50
1963    Euro    50
1963    Nort    5
1963    Ocea    23
1964    Afri    58
1964    Asia    51
1964    Cent    50
1964    Euro    50
1964    Nort    5
1964    Ocea    23
1965    Afri    58
1965    Asia    51
1965    Cent    50
1965    Euro    50
1965    Nort    5
1965    Ocea    23
1966    Afri    58
1966    Asia    51
1966    Cent    50
1966    Euro    50
1966    Nort    5
1966    Ocea    23
1967    Afri    58
1967    Asia    51
1967    Cent    50
```

```
1967    Euro    50
1967    Nort    5
1967    Ocea    23
1968    Afri    58
1968    Asia    51
1968    Cent    50
1968    Euro    50
1968    Nort    5
1968    Ocea    23
1969    Afri    58
1969    Asia    51
1969    Cent    50
1969    Euro    50
1969    Nort    5
1969    Ocea    23
1970    Afri    58
1970    Asia    51
1970    Cent    50
1970    Euro    50
1970    Nort    5
1970    Ocea    23
1971    Afri    58
1971    Asia    51
1971    Cent    50
1971    Euro    50
1971    Nort    5
1971    Ocea    23
1972    Afri    58
1972    Asia    51
1972    Cent    50
1972    Euro    50
1972    Nort    5
1972    Ocea    23
1973    Afri    58
1973    Asia    51
1973    Cent    50
1973    Euro    50
1973    Nort    5
1973    Ocea    23
1974    Afri    58
1974    Asia    51
1974    Cent    50
1974    Euro    50
1974    Nort    5
1974    Ocea    23
1975    Afri    58
1975    Asia    51
1975    Cent    50
```

```
1975    Euro    50
1975    Nort    5
1975    Ocea    23
1976    Afri    58
1976    Asia    51
1976    Cent    50
1976    Euro    50
1976    Nort    5
1976    Ocea    23
1977    Afri    58
1977    Asia    51
1977    Cent    50
1977    Euro    50
1977    Nort    5
1977    Ocea    23
1978    Afri    58
1978    Asia    51
1978    Cent    50
1978    Euro    50
1978    Nort    5
1978    Ocea    23
1979    Afri    58
1979    Asia    51
1979    Cent    50
1979    Euro    50
1979    Nort    5
1979    Ocea    23
1980    Afri    58
1980    Asia    51
1980    Cent    50
1980    Euro    50
1980    Nort    5
1980    Ocea    23
1981    Afri    58
1981    Asia    51
1981    Cent    50
1981    Euro    50
1981    Nort    5
1981    Ocea    23
1982    Afri    58
1982    Asia    51
1982    Cent    50
1982    Euro    50
1982    Nort    5
1982    Ocea    23
1983    Afri    58
1983    Asia    51
1983    Cent    50
```

```
1983    Euro    50
1983    Nort    5
1983    Ocea    23
1984    Afri    58
1984    Asia    51
1984    Cent    50
1984    Euro    50
1984    Nort    5
1984    Ocea    23
1985    Afri    58
1985    Asia    51
1985    Cent    50
1985    Euro    50
1985    Nort    5
1985    Ocea    23
1986    Afri    58
1986    Asia    51
1986    Cent    50
1986    Euro    50
1986    Nort    5
1986    Ocea    23
1987    Afri    58
1987    Asia    51
1987    Cent    50
1987    Euro    50
1987    Nort    5
1987    Ocea    23
1988    Afri    58
1988    Asia    51
1988    Cent    50
1988    Euro    50
1988    Nort    5
1988    Ocea    23
1989    Afri    58
1989    Asia    51
1989    Cent    50
1989    Euro    50
1989    Nort    5
1989    Ocea    23
1990    Afri    58
1990    Asia    51
1990    Cent    50
1990    Euro    50
1990    Nort    5
1990    Ocea    23
1991    Afri    58
1991    Asia    51
1991    Cent    50
```

```
1991    Euro    50
1991    Nort    5
1991    Ocea    23
1992    Afri    58
1992    Asia    51
1992    Cent    50
1992    Euro    50
1992    Nort    5
1992    Ocea    23
1993    Afri    58
1993    Asia    51
1993    Cent    50
1993    Euro    50
1993    Nort    5
1993    Ocea    23
1994    Afri    58
1994    Asia    51
1994    Cent    50
1994    Euro    50
1994    Nort    5
1994    Ocea    23
1995    Afri    58
1995    Asia    51
1995    Cent    50
1995    Euro    50
1995    Nort    5
1995    Ocea    23
1996    Afri    58
1996    Asia    51
1996    Cent    50
1996    Euro    50
1996    Nort    5
1996    Ocea    23
1997    Afri    58
1997    Asia    51
1997    Cent    50
1997    Euro    50
1997    Nort    5
1997    Ocea    23
1998    Afri    58
1998    Asia    51
1998    Cent    50
1998    Euro    50
1998    Nort    5
1998    Ocea    23
1999    Afri    58
1999    Asia    51
1999    Cent    50
```

```
1999    Euro    50
1999    Nort    5
1999    Ocea    23
2000    Afri    58
2000    Asia    51
2000    Cent    50
2000    Euro    50
2000    Nort    5
2000    Ocea    23
2001    Afri    58
2001    Asia    51
2001    Cent    50
2001    Euro    50
2001    Nort    5
2001    Ocea    23
2002    Afri    58
2002    Asia    51
2002    Cent    50
2002    Euro    50
2002    Nort    5
2002    Ocea    23
2003    Afri    58
2003    Asia    51
2003    Cent    50
2003    Euro    50
2003    Nort    5
2003    Ocea    23
2004    Afri    58
2004    Asia    51
2004    Cent    50
2004    Euro    50
2004    Nort    5
2004    Ocea    23
2005    Afri    58
2005    Asia    51
2005    Cent    50
2005    Euro    50
2005    Nort    5
2005    Ocea    23
2006    Afri    58
2006    Asia    51
2006    Cent    50
2006    Euro    50
2006    Nort    5
2006    Ocea    23
2007    Afri    58
2007    Asia    51
2007    Cent    50
```

```
2007    Euro    50
2007    Nort    5
2007    Ocea    23
2008    Afri    58
2008    Asia    51
2008    Cent    50
2008    Euro    50
2008    Nort    5
2008    Ocea    23
2009    Afri    58
2009    Asia    51
2009    Cent    50
2009    Euro    50
2009    Nort    5
2009    Ocea    23
2010    Afri    58
2010    Asia    51
2010    Cent    50
2010    Euro    50
2010    Nort    5
2010    Ocea    23
2011    Afri    58
2011    Asia    51
2011    Cent    50
2011    Euro    50
2011    Nort    5
2011    Ocea    23
2012    Afri    58
2012    Asia    51
2012    Cent    50
2012    Euro    50
2012    Nort    5
2012    Ocea    23
2013    Afri    58
2013    Asia    51
2013    Cent    50
2013    Euro    50
2013    Nort    5
2013    Ocea    23
2014    Afri    58
2014    Asia    51
2014    Cent    50
2014    Euro    50
2014    Nort    5
2014    Ocea    23
2015    Afri    58
2015    Asia    51
2015    Cent    50
```

```
2015    Euro    50
2015    Nort    5
2015    Ocea    23
2016    Afri    58
2016    Asia    51
2016    Cent    50
2016    Euro    50
2016    Nort    5
2016    Ocea    23
2017    Afri    58
2017    Asia    51
2017    Cent    50
2017    Euro    50
2017    Nort    5
2017    Ocea    23
2018    Afri    58
2018    Asia    51
2018    Cent    50
2018    Euro    50
2018    Nort    5
2018    Ocea    23
2019    Afri    58
2019    Asia    51
2019    Cent    50
2019    Euro    50
2019    Nort    5
2019    Ocea    23
2020    Afri    58
2020    Asia    51
2020    Cent    50
2020    Euro    50
2020    Nort    5
2020    Ocea    23
2021    Afri    58
2021    Asia    51
2021    Cent    50
2021    Euro    50
2021    Nort    5
2021    Ocea    23
2022    Afri    58
2022    Asia    51
2022    Cent    50
2022    Euro    50
2022    Nort    5
2022    Ocea    23
```

[66]:

```
# now combine this with stats
```

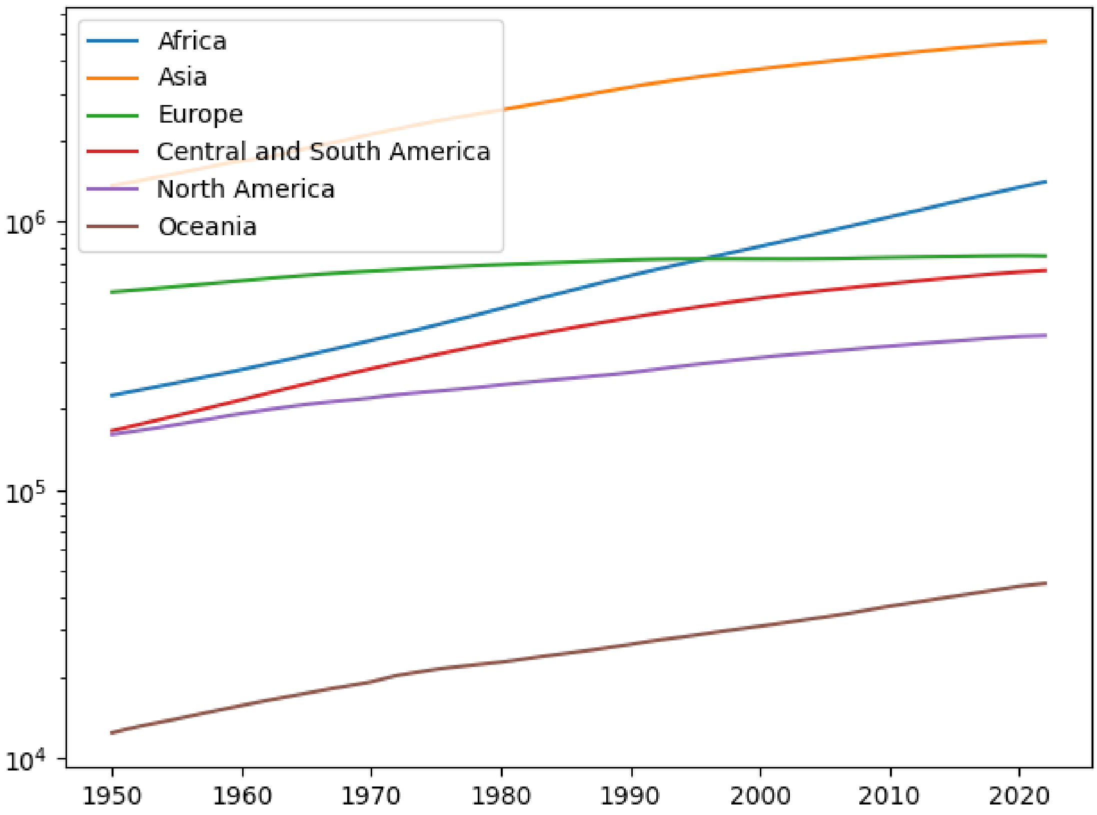
[44]:

```
dataGrouped=data.groupby(["Continent","Time"])
dataSummed=dataGrouped["TPopulation1Jan"].sum()
print(dataSummed)
```

```
Continent  Time
Africa     1950    225120.311
           1951    229978.205
           1952    234989.784
           1953    240182.336
           1954    245492.559
                      …
Oceania    2018     42175.314
           2019     42904.041
           2020     43652.259
           2021     44214.592
           2022     44768.856
Name: TPopulation1Jan, Length: 438, dtype: float64
```

[45]:

```
for cont in continents:
    plt.plot(dataSummed[cont].index,dataSummed[cont].values,label=cont)
plt.yscale("log")
plt.legend()
plt.tight_layout()
plt.show()
```


Africa
Asia
Europe
Central and South America
North America
Oceania
10^6
10^5
10^4
1950
1960
1970
1980
1990
2000
2010
2020

[ ]:

[ ]: