Getting Started in Polars

Alier Reng

2024-04-20

Motivation

Getting Started

Loading the Required Libraries

```
# Libraries -------
import polars as pl
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Importing Data

```
census_raw = pl.read_csv("data/ss_2008_census_data_raw.csv", null_values="NA")

# Inspect the first 5 rows
print(census_raw.head(5))
```

```
shape: (5, 10)
```

Region	Region Name	Region - RegionId	Variable	…	Age Name	Scale	Units	2008
str	str	str	str		str	str	str	i64
KN.A2	Upper Nile	SS-NU	KN.B2	…	Total	units	Persons	964353

```
KN.A2   Upper Nile   SS-NU            KN.B2     …   0 to 4    units   Persons   150872
KN.A2   Upper Nile   SS-NU            KN.B2     …   5 to 9    units   Persons   151467
KN.A2   Upper Nile   SS-NU            KN.B2     …   10 to 14  units   Persons   126140
KN.A2   Upper Nile   SS-NU            KN.B2     …   15 to 19  units   Persons   103804
```

Importing the Data Lazily

```
census_lazy = pl.scan_csv(
    'data/ss_2008_census_data_raw.csv', null_values='NA'
    )

# Inspect the first 5 rows
print(census_lazy.collect().head(5))
```

```
shape: (5, 10)
```

Region	Region Name	Region - RegionId	Variable	…	Age Name	Scale	Units	2008
str	str	str	str		str	str	str	i64
KN.A2	Upper Nile	SS-NU	KN.B2	…	Total	units	Persons	964353
KN.A2	Upper Nile	SS-NU	KN.B2	…	0 to 4	units	Persons	150872
KN.A2	Upper Nile	SS-NU	KN.B2	…	5 to 9	units	Persons	151467
KN.A2	Upper Nile	SS-NU	KN.B2	…	10 to 14	units	Persons	126140
KN.A2	Upper Nile	SS-NU	KN.B2	…	15 to 19	units	Persons	103804

```
# Inspect the last 5 rows
print(census_raw.tail(5))
```

```
shape: (5, 10)
```

Region	Region Name	Region - RegionId	Variable	…	Age Name	Scale	Units	200
str	str	str	str		str	str	str	i64
KN.A11	Eastern Equatoria	SS-EE	KN.B8	…	60 to 64	units	Persons	527

KN.A11	Eastern Equatoria	SS-EE	KN.B8	…	65+	units	Persons	863
null	null	null	null	…	null	null	null	nul
Source:	National Bureau of Statistics, S…	null	null	…	null	null	null	nul
Download URL:	http://southsudan.opendataforafr…	null	null	…	null	null	null	nul

```
# Inspect random 5 rows
print(census_raw.sample(5))
```

```
shape: (5, 10)
```

Region	Region Name	Region - RegionId	Variable	…	Age Name	Scale	Units	2008
str	str	str	str		str	str	str	i64
KN.A4	Unity	SS-UY	KN.B2	…	30 to 34	units	Persons	34200
KN.A4	Unity	SS-UY	KN.B5	…	60 to 64	units	Persons	4160
KN.A4	Unity	SS-UY	KN.B2	…	15 to 19	units	Persons	59342
KN.A11	Eastern Equatoria	SS-EE	KN.B8	…	35 to 39	units	Persons	25808
KN.A9	Western Equatoria	SS-EW	KN.B5	…	20 to 24	units	Persons	29084

Checking for Missing Values

```
# Inspect the last 5 rows
print(census_raw.null_count())
```

```
shape: (1, 10)
```

Region	Region Name	Region - RegionId	Variable	…	Age Name	Scale	Units	2008
u32	u32	u32	u32		u32	u32	u32	u32
1	1	3	3	…	3	3	3	3

```
import polars.selectors as cs
# Inspect the last 5 rows
print(
    census_raw
    .select(cs.all().is_null().sum())
)
```

```
shape: (1, 10)
```

Region	Region Name	Region - RegionId	Variable	…	Age Name	Scale	Units	2008
u32	u32	u32	u32		u32	u32	u32	u32
1	1	3	3	…	3	3	3	3

Selecting Columns of Interest

```
# Selecting columns of interest: polars provides various methods for selecting columns
print(
    census_raw
    .select(cs.ends_with('Name'), '2008')
    .columns
)
```

```
['Region Name', 'Variable Name', 'Age Name', '2008']
```

```
age_mapping = {
    "0 to 4": "0-14",
    "5 to 9": "0-14",
    "10 to 14": "0-14",
    "15 to 19": "15-24",
    "20 to 24": "15-24",
    "25 to 29": "25-34",
    "30 to 34": "25-34",
    "35 to 39": "35-44",
    "40 to 44": "35-44",
    "45 to 49": "45-54",
    "50 to 54": "45-54",
```

```
    "55 to 59": "55-64",
    "60 to 64": "55-64",
    "65+": "65 and above",
}
census = (
    census_raw
    .select(
        ["Region Name", "Variable Name", "Age Name", "2008"]
    )
    .rename(
        {
            "Region Name": "state",
            "Variable Name": "gender",
            "Age Name": "age_category",
            "2008": "population",
        }
    )
    .with_columns(
        gender=pl.col("gender").str.split(" ").list.get(1),
        age_category=pl.col("age_category").replace(age_mapping),
    )
    .filter(
        (pl.col("gender") != "Total") & (pl.col("age_category") != "Total")
    )
    .group_by(['state', 'gender', 'age_category'])
    .agg(total=pl.col('population').sum())
    .sort('total', descending=True)
)
```

```
print(census.head(5))
```

```
shape: (5, 4)
```

state	gender	age_category	total
str	str	str	i64
Jonglei	Male	0-14	338443

Jonglei	Female	0-14	263646
Central Equatoria	Male	0-14	242247
Upper Nile	Male	0-14	237461
Warrap	Male	0-14	230854