Comprehensive Guide

Data Reshaping and Organization with Polars

Practical Guide for Working with Polars

2024

Table of Contents

Chapter 1: Introduction and Setup

What is Polars?

Polars is an extremely fast data processing library written in Rust, with a convenient Python API. It provides excellent performance and efficient memory usage.

Advantages of Polars:

Speed	Up to 10x faster than Pandas
Memory Efficiency	Smart memory usage
Lazy Evaluation	Automatic optimization
Clean API	Intuitive syntax

Installation

```
pip install polars
```

Basic Import

```
import polars as pl
from polars import selectors as cs
```

Chapter 2: Unpivot - Columns to Rows

What is Unpivot?

Unpivot (or Melt) is used to transform data from 'wide' format to 'long' format. This is useful when you want to analyze or visualize data by category.

Basic Svntax

```
df.unpivot(
index='academic_year',
on=['students', 'graduate'],
variable_name='student_type',
value_name='count'
)
```

Using Selectors

```
df.unpivot(
index='academic_year',
on=cs.numeric() # All numeric columns
)
```

Chapter 3: Pivot - Rows to Columns

What is Pivot?

Pivot is the inverse operation of Unpivot. It transforms data from 'long' format to 'wide' format. This is useful when you want to see comparisons between categories or create an organized table.

Basic Syntax

```
df.pivot(
index='academic_year',
values='count',
on='student_type'
)
```

Handling Duplicates

When there are duplicates in the data, you must specify an aggregation function:

```
df.pivot(
index='year',
values='count',
on='category',
aggregate_function='sum'
)
```

Common Aggregation Functions

Function	Description
'sum'	Sum
'mean'	Average
'min'	Minimum
'max'	Maximum
'count'	Count
pl.element()	List of all values

Chapter 4: Join - Combining Tables

What is Join?

Join allows combining two tables based on a common key column. This is useful when you have complementary information in different tables.

Basic Syntax

```
df1.join(
df2,
left_on='academic_year',
right_on='year',
how='inner'
)
```

Types of Joins

Type	Description	When to Use
inner	Only common rows	Shared data only
left	All rows from left table	Keep all data from table 1
right	All rows from right table	Keep all data from table 2
outer	All rows from both tables	All data

Validation

You can verify that the join was performed correctly using the validate parameter:

```
df1.join(df2, on='id', how='inner', validate='1:m')
```

Chapter 5: Concatenation - Stacking Tables

Vertical Concatenation

Adding rows below each other:

```
pl.concat([df1, df2, df3], how='vertical')
```

Horizontal Concatenation

Adding columns side by side:

```
pl.concat([df1, df2], how='horizontal')
```

Additional Methods

Method	Description	Example
concat	Flexible, multiple DataFrames	pl.concat([df1, df2])
vstack	Fast, two DataFrames	df1.vstack(df2)
hstack	Horizontal, requires equal lengths	df1.hstack(df2)
extend	Modifies the source	df1.extend(df2)

Chapter 6: Additional Techniques

Partition - Dividing into Groups

```
partitions = df.partition_by('category')
```

Transpose - Flipping Rows and Columns

```
df.transpose(include_header=True)
```

Reshape - Changing Shape

```
df.select(pl.col('column').reshape((rows, cols)))
```

Chapter 7: Selectors - Smart Selection

Common Selectors

Selector	Description
cs.numeric()	All numbers
cs.string()	All strings
cs.float()	Float only
cs.integer()	Integer only
cs.starts_with('x')	Starts with x
cs.ends_with('x')	Ends with x
cs.contains('x')	Contains x

Chapter 8: Tips and Debugging

Performance Tips

- Use LazyFrame whenever possible
- Filter data early before heavy operations
- Use Selectors for dynamic selection
- Check data size before operations

Common Errors

Error	Solution
multiple elements	Add aggregate_function
validation failed	Change validate or fix data
shape mismatch	Use concat instead of hstack

Chapter 9: Summary and Resources

Summary

In this guide we learned all the important techniques for reshaping and organizing data in Polars: Unpivot, Pivot, Join, Concatenation and additional techniques. Polars provides powerful and efficient tools for working with data.

Additional Resources

- Official Documentation: https://pola-rs.github.io/polars/
- Discord Community: https://discord.gg/4UfP5cfBE7
- GitHub: https://github.com/pola-rs/polars
- Videos: YouTube - Polars Official

Good luck working with Polars! ■