Transforming and Cleaning Unstructured Data

Swetha Kolalapudi
CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Transforming data using functional constructs i.e. filter, map and reduce

Cleaning unstructured data

Identifying and removing anomalies, missing values

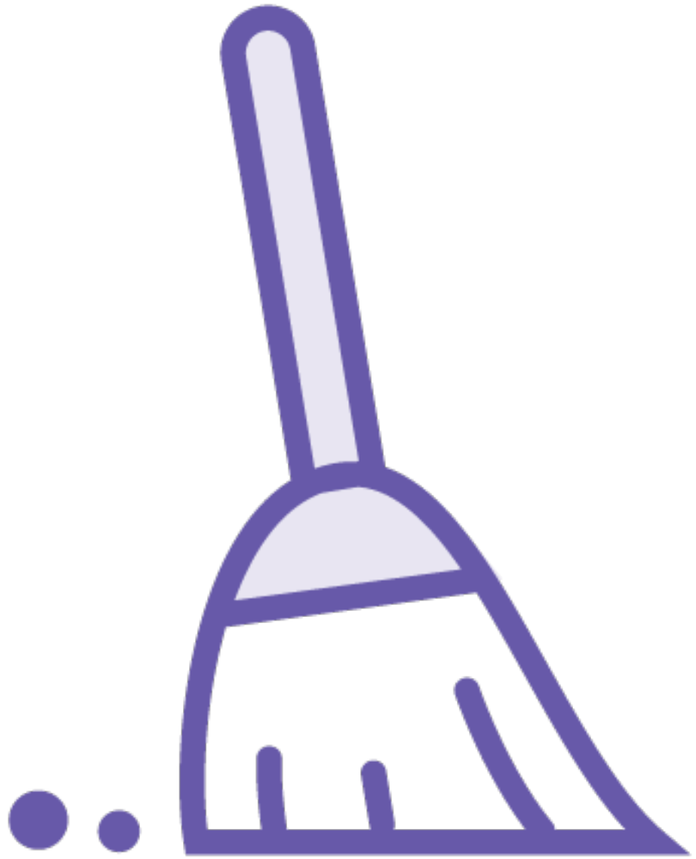
Crime in New York City

Drawing Insights

What is the trend in crime over the past few years?

Which categories of crimes are the most common?

In which boroughs is a particular category of crime most prevalent?

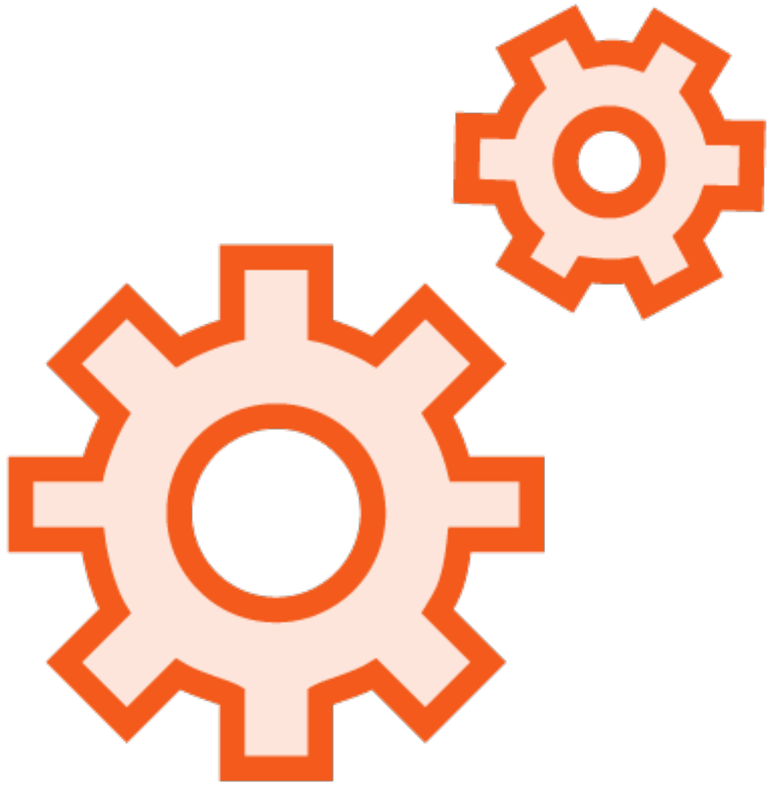


Cleaning Data

Filtering the header

Missing values

Anomalous data



Transforming Data

Extracting fields

Computing metrics

Demo

Getting a first sense of the data

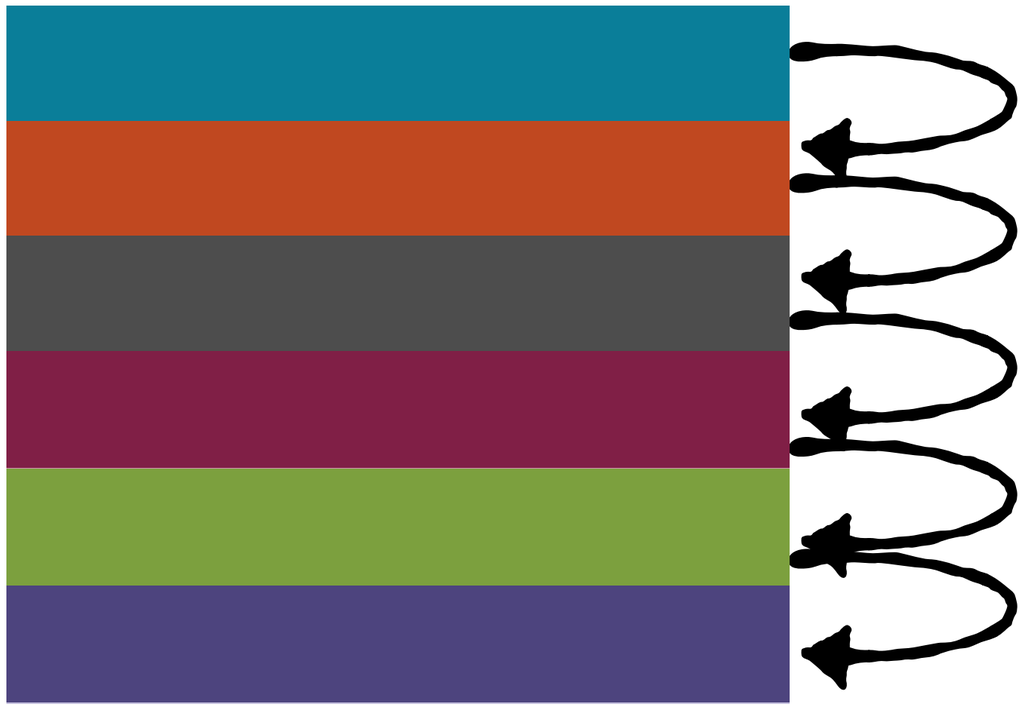
Transforming Data with Spark

An RDD is a collection of records

Transforming Data with Spark

How do you do something with a collection of objects?

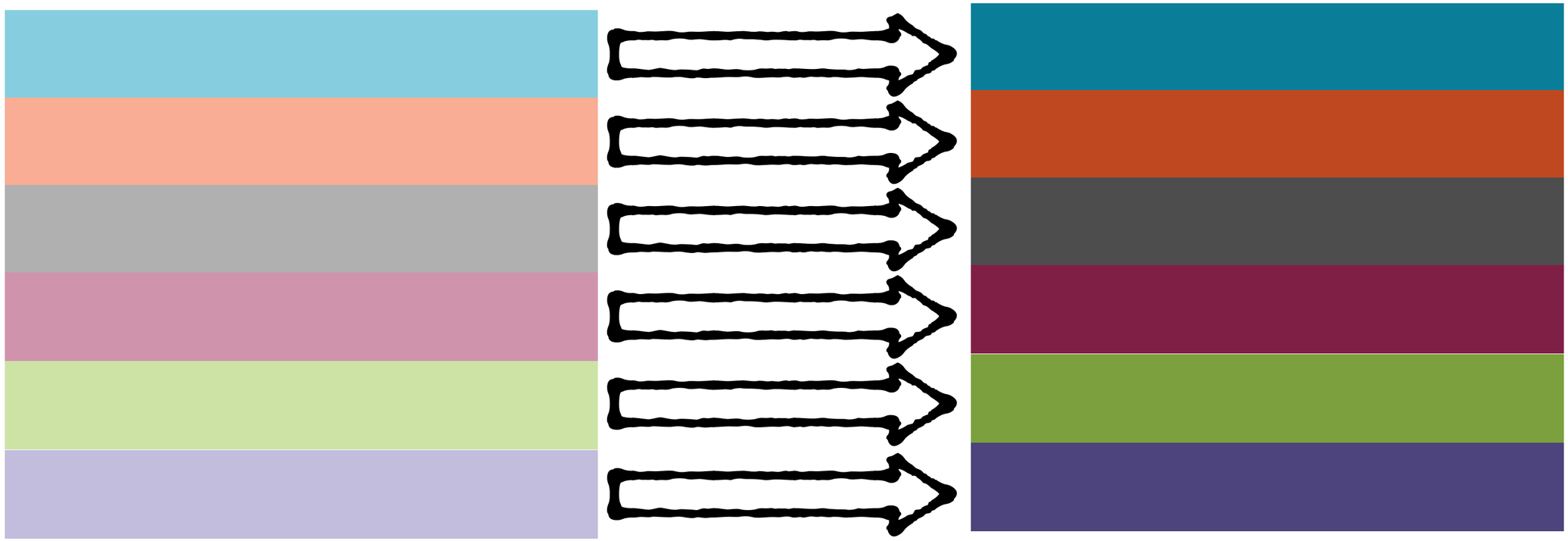
Transforming Data with Spark



Imperative way

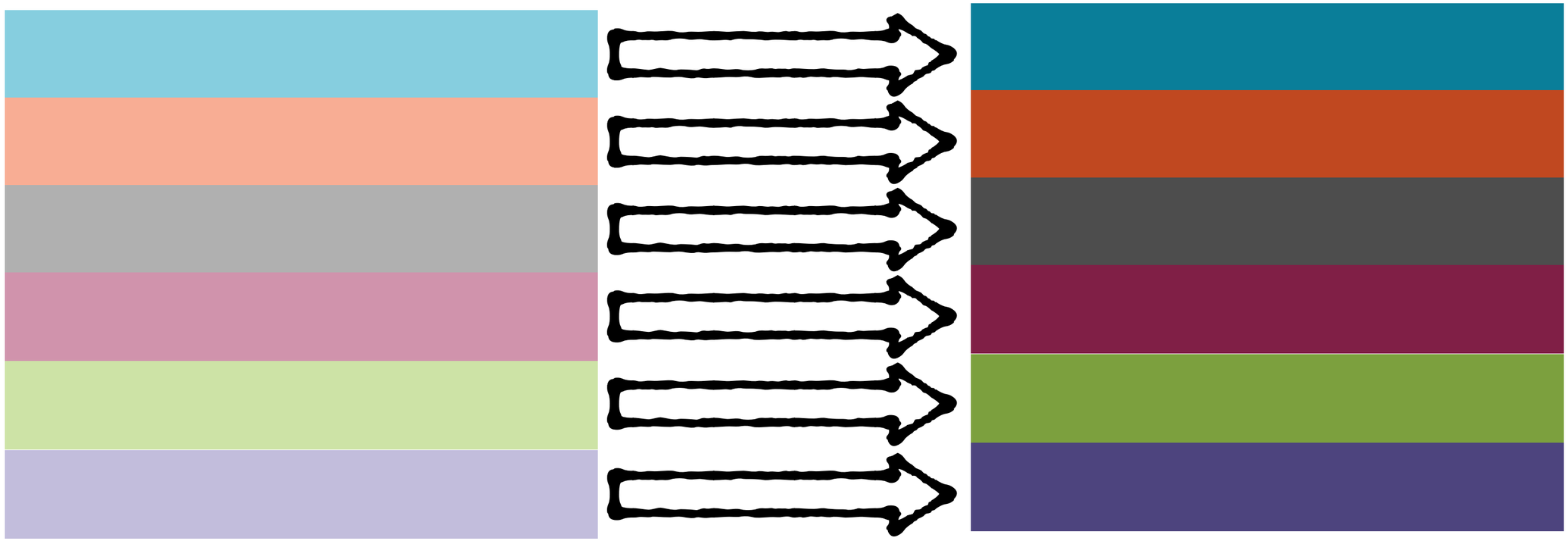
Using loops

Transforming Data with Spark



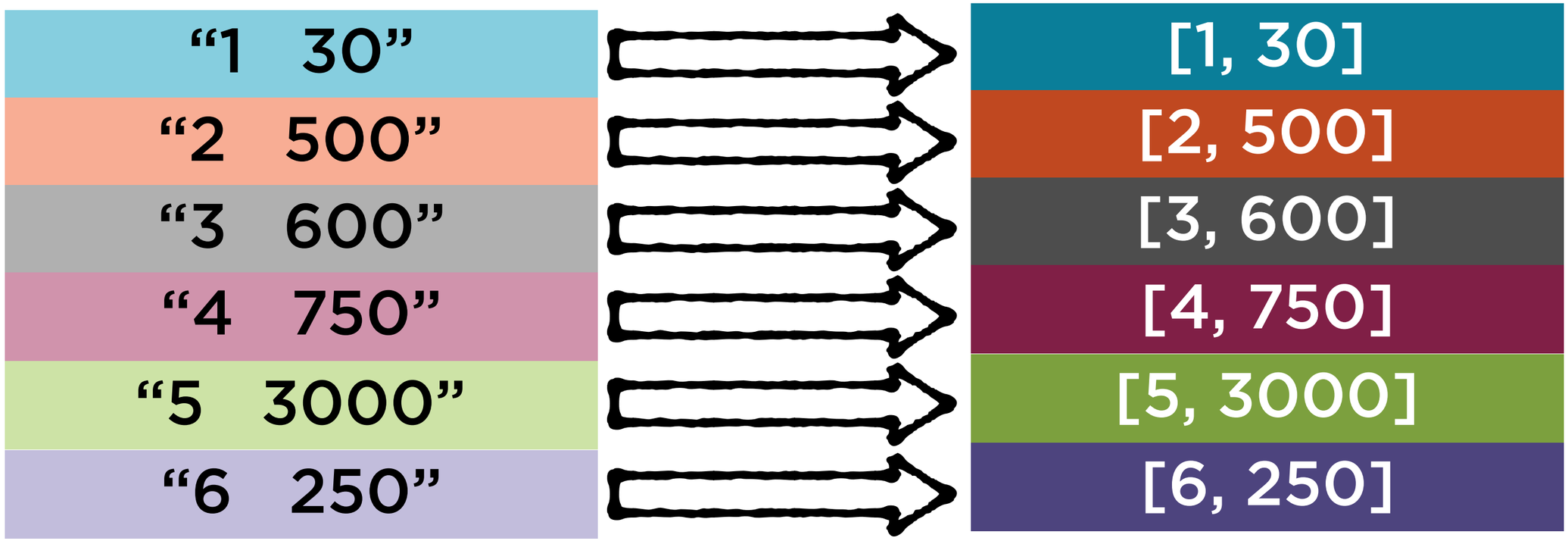
Functional way

Transforming Data with Spark



Apply the same function to each record

Transforming Data with Spark



Split and parse a record

Functional programming
allows you to process data
in parallel

Explicitly Defined Functions

```
def parse(row):
    reader = csv.reader(StringIO(row))
    row=reader.next()
    return Crime(*row)
```

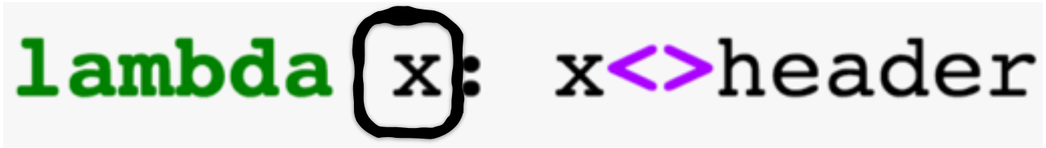
Input is an RDD record

Lambda Functions

```
lambda x: x<>header
```

Anonymous functions
One time use

Lambda Functions



Input

Lambda Functions

```
lambda x: x<>header
```

Output

Functional Programming

Filter records matching a condition

Transform each record to another record

Combine records in a specified way

filter

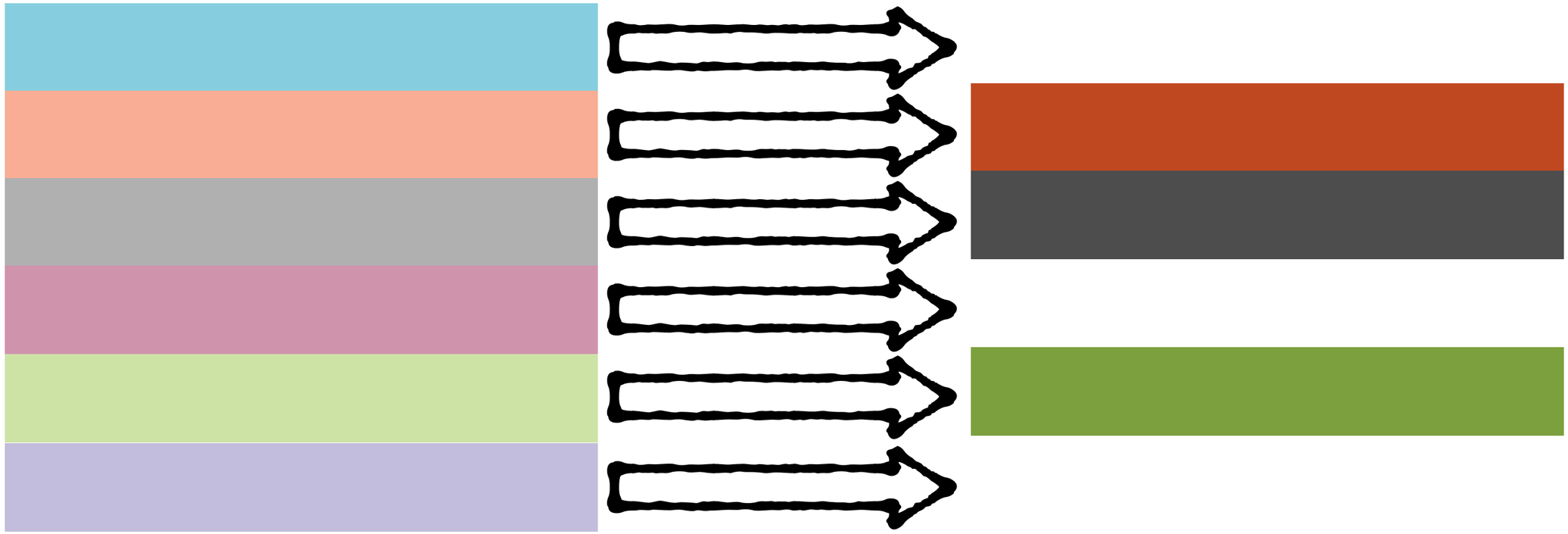
Filter records matching a given condition

filter

Apply a boolean function

True	Keep
False	Drop

Filtering Records



Functional Programming

Filter records matching a condition

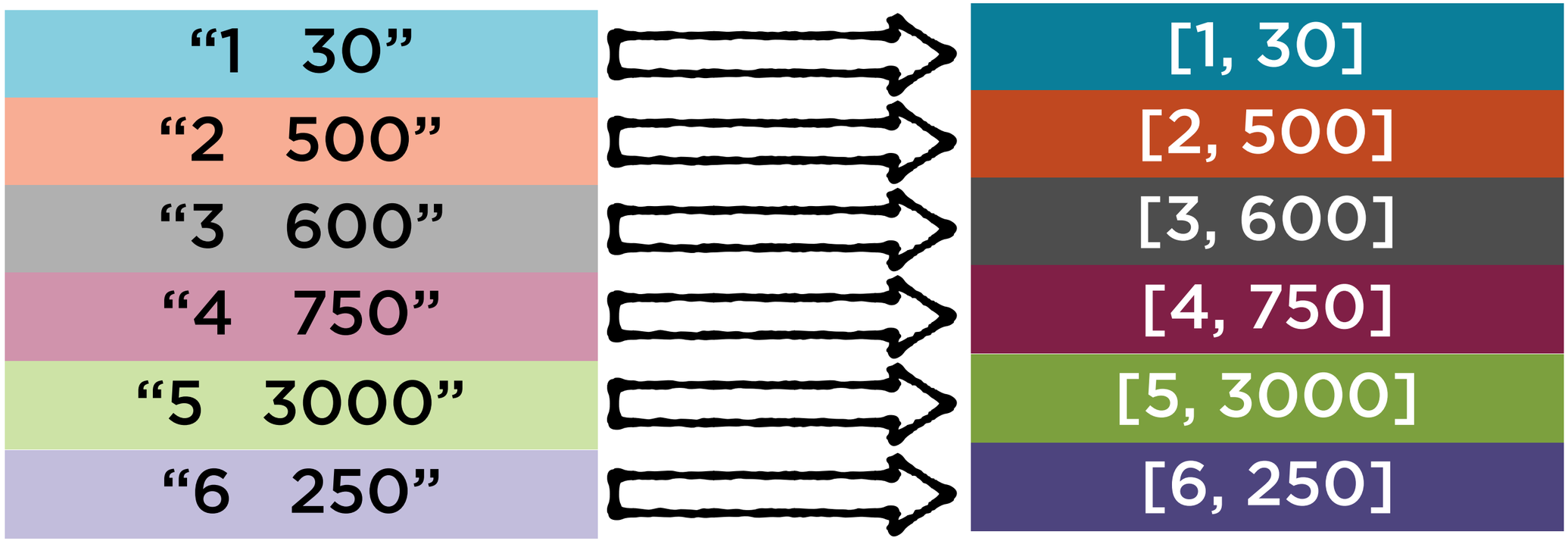
Transform each record to another record

Combine records in a specified way

map

Transform a record to another record

Transforming Records



Split and parse a record

Functional Programming

Filter records matching a condition

Transform each record to another record

Combine records in a specified way

reduce

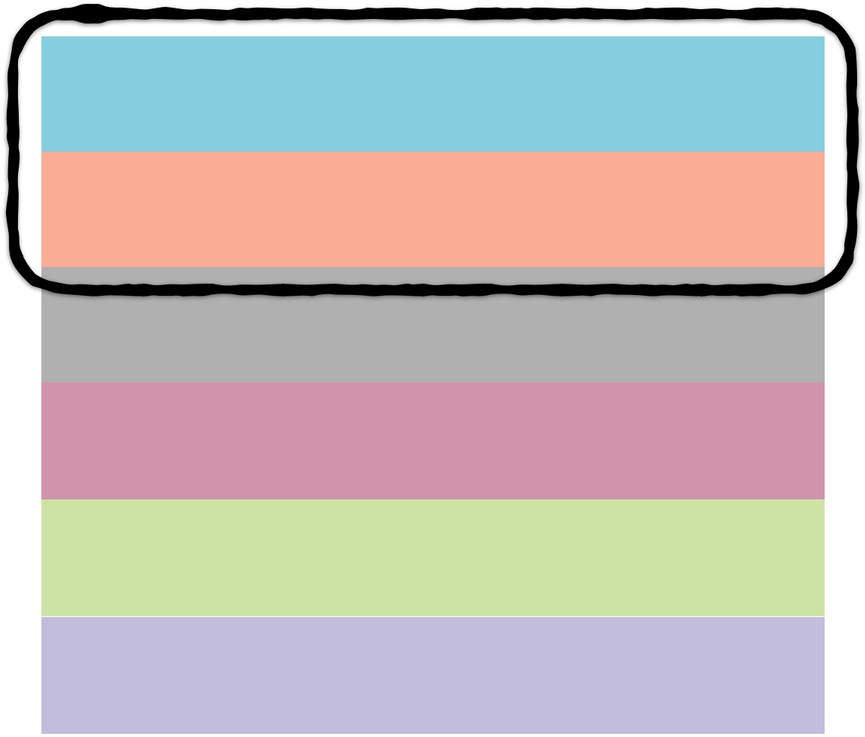
Combine records in a specified way

Sum

Maximum

Minimum

Combining Records



A function with 2 arguments

Combining Records

A function with 2 arguments

Combining Records

A function with 2 arguments

Combining Records

A function with 2 arguments

Combining Records

A function with 2 arguments

Combining Records

A function with 2 arguments

Functional Programming

Filter records matching a condition

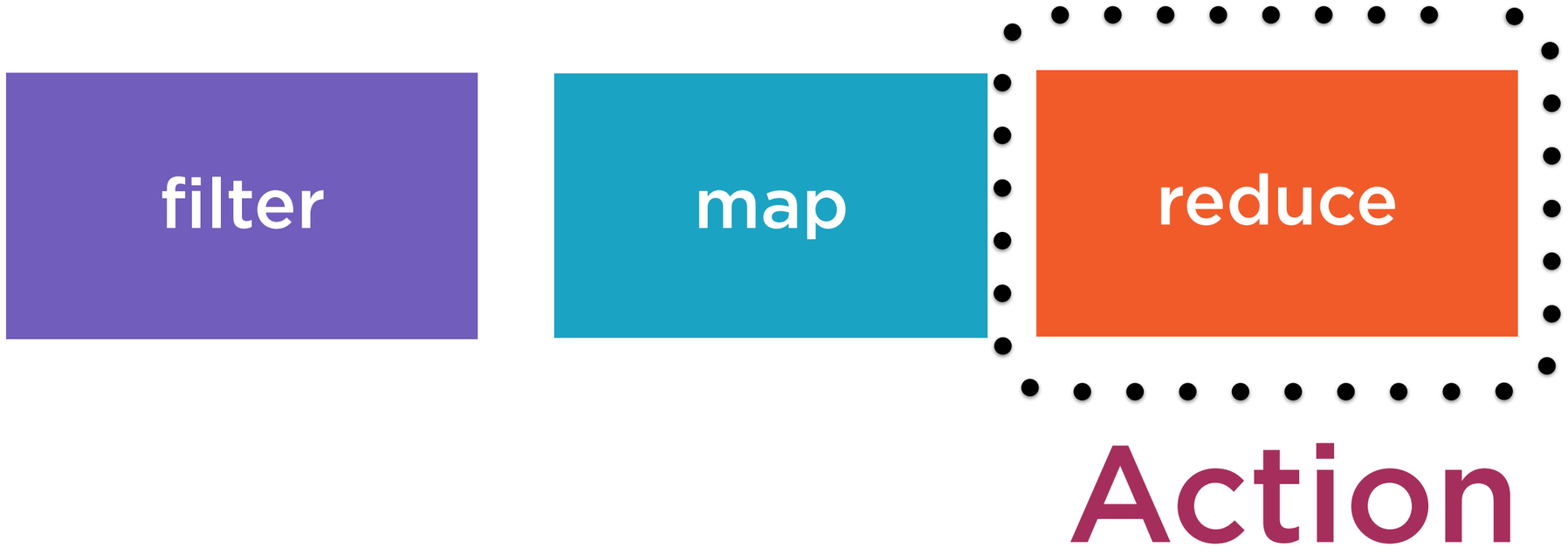
Transform each record to another record

Combine records in a specified way

Functional Programming in Spark

Transformations

Functional Programming in Spark



Demo

Filtering the header row

Demo

Transforming records from strings to named tuples

Demo

Identifying missing values

Filtering records with missing values

Demo

Identifying anomalies

Filtering records with anomalies

Demo

Drawing insights from New York City Crime data

Summary

Filter records matching a certain condition

Transform data using the map operation

Compute aggregates using the reduce operation

Identify and remove anomalies, missing values