Oberon Based Data Frame Tool

Balancing Simplicity and Performance

Irena Torosyan
Supervisor: Norayr Chilingarian
May, 2024

Capstone project submitted in fulfillment of requirements for the degree of BS in Data Science at the American University of Armenia

Related Work

A Landscape of Tabular Data Exploration Tools

- VisiCalc (1979)
- Spreadsheets (1980s)
- Statistical Programming Languages (1990s)
- Diversification and Specialization (present)

Image from “Apple II History”

Data Frame Design

Module Frame

- Naming Columns
- Column Types
- Loaders
  - e.g. frame.setLoader(f1, readcsv.ReadCSVFile);
- OOP Considerations
  - e.g. f1.load(f1, "country_full.csv", TRUE);

```
DEFINITION frame;

  TYPE
    Tloader = PROCEDURE (f: frm; filename: ARRAY OF CHAR; hasColumns: BOOLEAN);
    cell = POINTER TO cellDesc;
    cellDesc = RECORD [100H]
    END;
    cnames = POINTER TO ARRAY OF columnName;
    #column = POINTER TO ARRAY OF cell;
    columnName = ARRAY 32 OF CHAR;
    frm = POINTER TO frmDesc;
    frmDesc = RECORD
      height-: INT16;
      width-: INT16;
      mtrx: #matrix;
      columnNames-: cnames;
      hasColumnNames-: BOOLEAN;
      load-: Tloader;
    END;
    intCell = POINTER TO intCellDesc;
    intCellDesc = RECORD (cellDesc)
      int: INT32;
    END;
    #matrix = POINTER TO ARRAY OF #column;
    strCell = POINTER TO strCellDesc;
    strCellDesc = RECORD (cellDesc)
      string: ARRAY 256 OF CHAR;
    END;

  PROCEDURE create(): frm;
  PROCEDURE printColumnNames(f: frm);
  PROCEDURE printDataFrame(f: frm);
  PROCEDURE read(f: frm; c: INT16; r: INT16): cell;
  PROCEDURE setColNames(VAR f: frm; names: cnames);
  PROCEDURE setLoader(f: frm; l: Tloader);
  PROCEDURE setSize(VAR f: frm; w: INT16; h: INT16);
  PROCEDURE write(f: frm; cl: cell; c: INT16; r: INT16);

END frame.
```

Loaders & Writers

Format Agnostic Data Handling

- Automatic Schema Discovery
- Column Name Handling
- Data Type Inference
- File Format Conversion

```
11x249
name alpha-2 alpha-3 country-code iso_3166-2 region sub-region intermediate-region region-code sub-region-code intermediate-region-code
Afghanistan AF AFG 4 ISO 3166-2:AF Asia Southern Asia "" 142 34 ""
Åland Islands AX ALA 248 ISO 3166-2:AX Europe Northern Europe "" 150 154 ""
Albania AL ALB 8 ISO 3166-2:AL Europe Southern Europe "" 150 39 ""
Algeria DZ DZA 12 ISO 3166-2:DZ Africa Northern Africa "" 2 15 ""
American Samoa AS ASM 16 ISO 3166-2:AS Oceania Polynesia "" 9 61 ""
Andorra AD AND 20 ISO 3166-2:AD Europe Southern Europe "" 150 39 ""
Angola AO AGO 24 ISO 3166-2:AO Africa Sub-Saharan Africa Middle Africa 2 202 17
Anguilla AI AIA 660 ISO 3166-2:AI Americas Latin America and the Caribbean Caribbean 19 419 29
Antarctica AQ ATA 10 ISO 3166-2:AQ "" "" "" "" "" ""
Antigua and Barbuda AG ATG 28 ISO 3166-2:AG Americas Latin America and the Caribbean Caribbean 19 419 29
Argentina AR ARG 32 ISO 3166-2:AR Americas Latin America and the Caribbean South America 19 419 5
Armenia AM ARM 51 ISO 3166-2:AM Asia Western Asia "" 142 145 ""
Aruba AW ABW 533 ISO 3166-2:AW Americas Latin America and the Caribbean Caribbean 19 419 29
Australia AU AUS 36 ISO 3166-2:AU Oceania Australia and New Zealand "" 9 53 ""
Austria AT AUT 40 ISO 3166-2:AT Europe Western Europe "" 150 155 ""
Azerbaijan AZ AZE 31 ISO 3166-2:AZ Asia Western Asia "" 142 145 ""
Bahamas BS BHS 44 ISO 3166-2:BS Americas Latin America and the Caribbean Caribbean 19 419 29
Bahrain BH BHR 48 ISO 3166-2:BH Asia Western Asia "" 142 145 ""
```

```
3x4
0 a NIL
1 b NIL
2 c NIL
3 d NIL
```

```
3x4
0 1 2
0 a NIL
1 b NIL
2 c NIL
3 d NIL
```

Statistical Analysis

Module Stats

- Column Wise Analysis
- Frame Wise Analysis
- Separate Statistics
- Combined Functionality

```
Column2
Column type = string
Number of NIL = 0
Number of Integers = 0
Number of Strings = 249
No integer values found

Column3
Column type = int
Number of NIL = 0
Number of Integers = 249
Number of Strings = 0
Sum = 108025
Mean = 4.33835E+02
Minimum Value = 4
Maximum Value = 894
Median = 4.34E+02
Mode = 4
Frequency of mode = 1
```

Performance Analysis

Frame VS Pandas VS csvkit

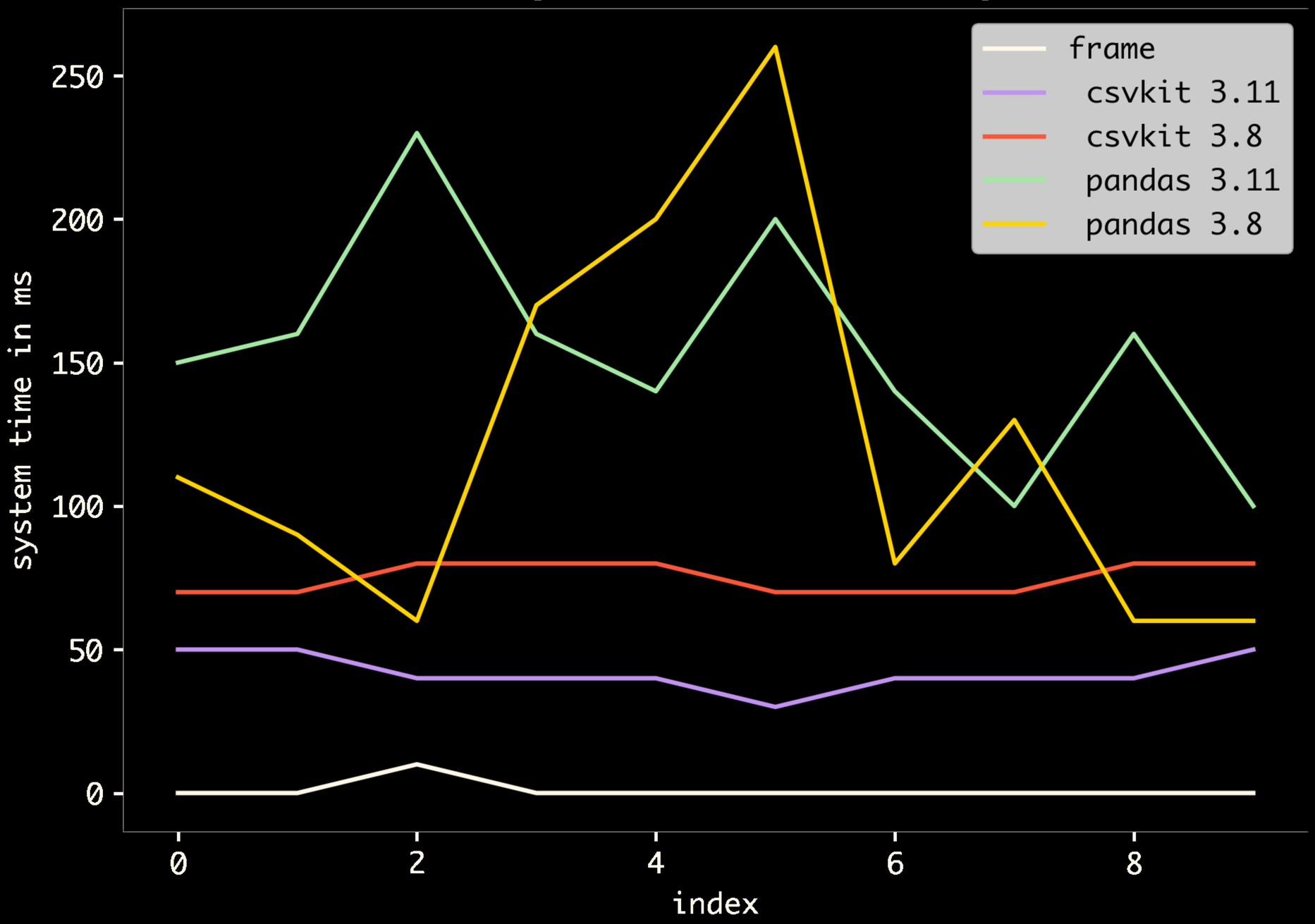
- country_full.csv
  - A relatively small file with 11 columns and around 250 rows.
- people-10000.csv
  - A larger dataset with 5 columns and around 10000 rows.

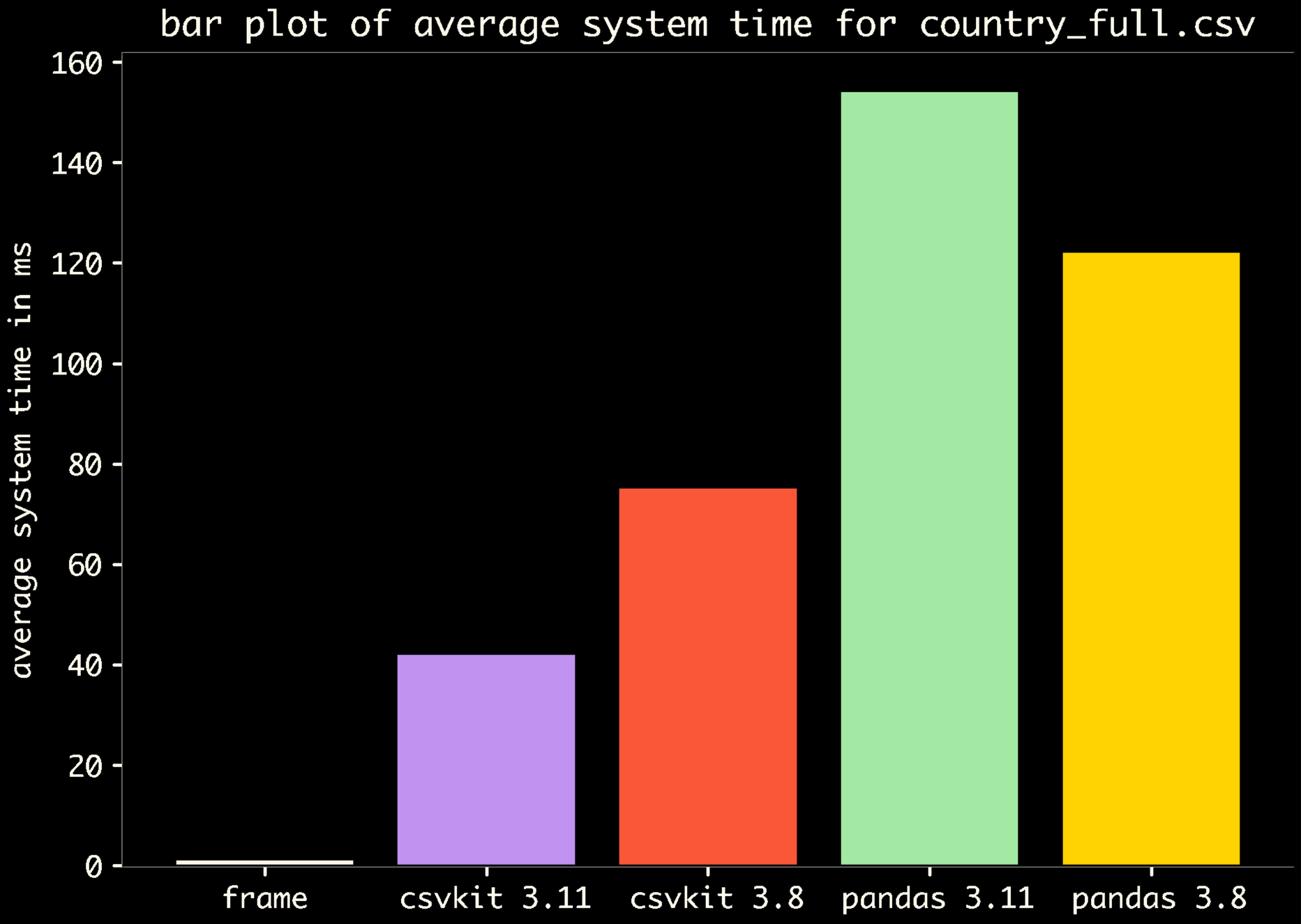
Average System Time

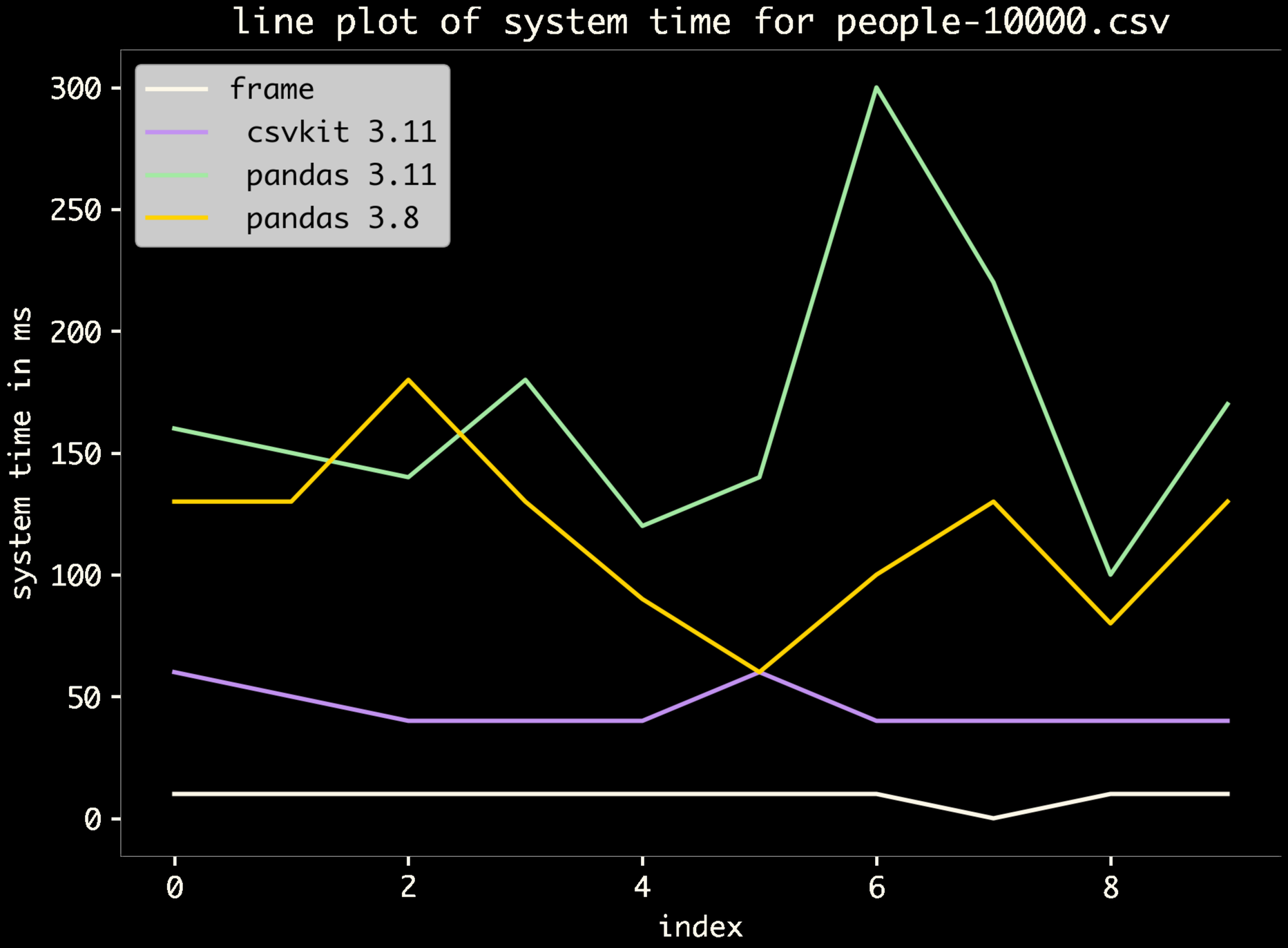
country_full.csv VS people-10000.csv

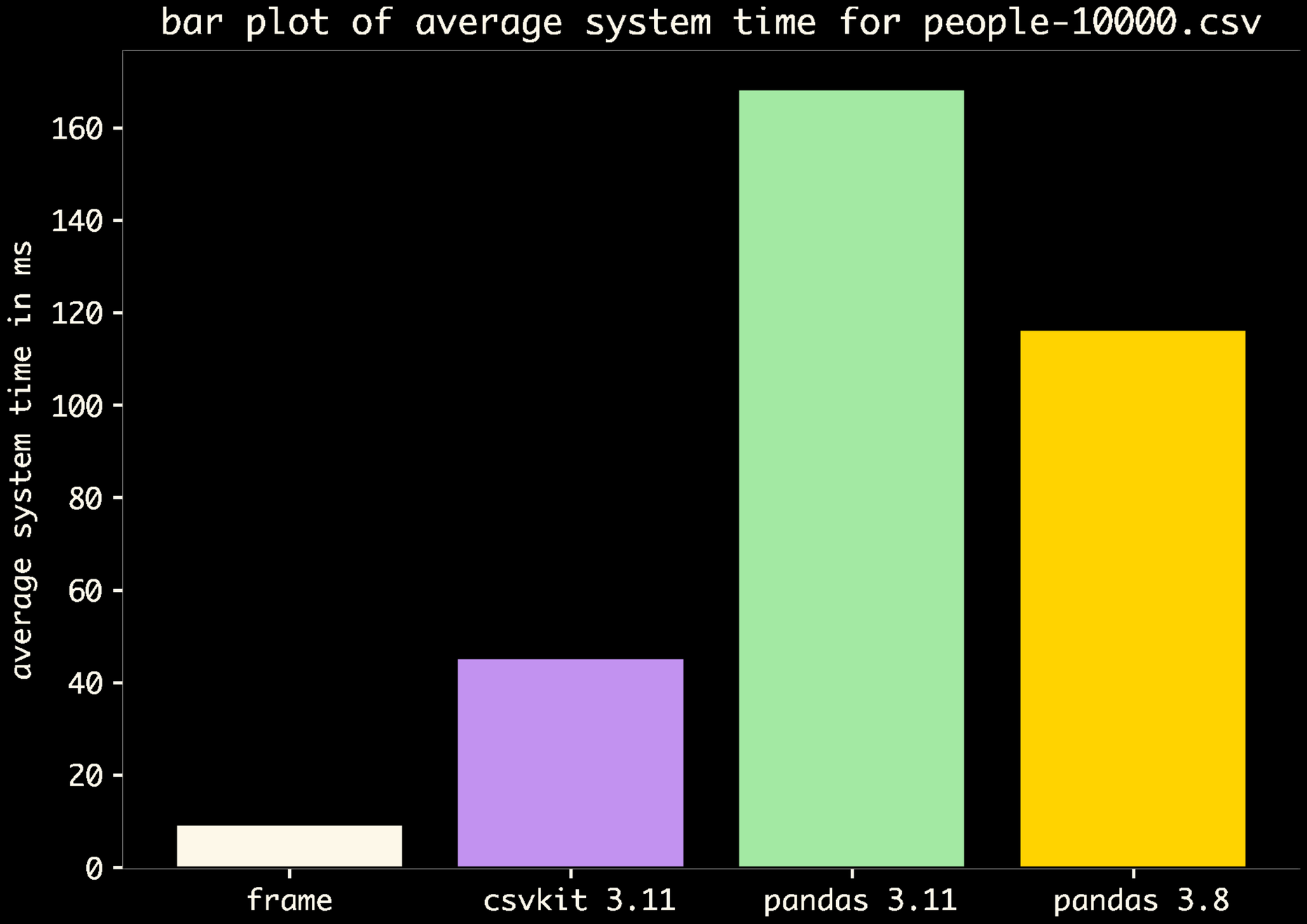
```
 frame
sum = 10
mean = 1.0E+00

 csvkit 3.11
sum = 420
mean = 4.2E+01

 csvkit 3.8
sum = 750
mean = 7.5E+01

 pandas 3.11
sum = 1540
mean = 1.54E+02

 pandas 3.8
sum = 1220
mean = 1.22E+02
```

```
 frame
sum = 90
mean = 9.0E+00

 csvkit 3.11
sum = 450
mean = 4.5E+01

 csvkit 3.8
No integer values found

 pandas 3.11
sum = 1680
mean = 1.68E+02

 pandas 3.8
sum = 1160
mean = 1.16E+02
```


line plot of system time for country_full.csv
system time in ms
250
200
150
100
50
0
0
2
4
6
8
index
frame
csvkit 3.11
csvkit 3.8
pandas 3.11
pandas 3.8


bar plot of average system time for country_full.csv
average system time in ms
0
20
40
60
80
100
120
140
160
frame
csvkit 3.11
csvkit 3.8
pandas 3.11
pandas 3.8


line plot of system time for people-10000.csv
frame
csvkit 3.11
pandas 3.11
pandas 3.8
system time in ms
300
250
200
150
100
50
0
0
2
4
6
8
index


bar plot of average system time for people-10000.csv
average system time in ms
0
20
40
60
80
100
120
140
160
frame
csvkit 3.11
pandas 3.11
pandas 3.8

What Comes After?
Future Development

- C code output
- Python Wrapper
- File Format Support
- Functionality Refinement
- Development with Oberon

```
/* voc 2.1.0 [2023/11/10] for clang LP64 on darwin xtpam */

#define SHORTINT INT8
#define INTEGER  INT16
#define LONGINT  INT32
#define SET      UINT32

#include "SYSTEM.h"
#include "oocIntStr.h"
#include "Out.h"
#include "Strings.h"
#include "frame.h"
#include "readcsv.h"
#include "stats.h"
#include "writecsv.h"




static void testStat_main (void);


static void testStat_main (void)
{
        frame_frm f = NIL;
        INT16 i;
        f = frame_create();
        frame_setLoader(f, readcsv_ReadCSVFile);
        (*f->load)(f, (CHAR*)"p_time_in_ms.csv", 17, 1);
        i = 0;
        Out_Char(' ');
        do {
                Out_String((f->columnNames->data)[__X(i, f->columnNames->len[0])], 32);
                Out_Ln();
                stats_mean(f, i);
                Out_Ln();
                i += 1;
        } while (!(i == f->width));
}


export int main(int argc, char **argv)
{
        __INIT(argc, argv);
        __MODULE_IMPORT(oocIntStr);
        __MODULE_IMPORT(Out);
        __MODULE_IMPORT(Strings);
        __MODULE_IMPORT(frame);
        __MODULE_IMPORT(readcsv);
        __MODULE_IMPORT(stats);
        __MODULE_IMPORT(writecsv);
        __REGMAIN("testStat", 0);
/* BEGIN */
        testStat_main();
        __FINI;
}
```

Thank You!