Exploratory Data Analysis in Pharo

Oleksandr Zaytsev
olk.zaytsev@gmail.com

Abstract

The simplicity and power of Smalltalk combined with a live environment provided by Pharo make up a perfect environment for data analysis. The advanced debugging and inspecting tools together with the library for agile visualizations allow us to communicate and play with every object in our system. This includes all the logical components of both data and the algorithm. Provided the proper tools and the open source libraries for machine learning, statistics, and optimization, Pharo can become both powerful tool for professional data analysts, and a simple environment for everyone who wants to play with a data set. Many important tools and algorithms are already implemented in PolyMath. But the essential part of data analysis toolkit is missing - the specialized data structures for structured data sets with a simple and powerful API for summarizing, cleaning, and manipulating the data. In this paper I introduce `DataFrame` and `DataSeries` - the new collections specifically designed for working with structured data. I demonstrate how these tools can be used for descriptive statistics and the exploratory data analysis - the critical first step of data analysis which allows us to get the summary of a data set, detect mistakes, determine the relations, and select the appropriate model for further confirmatory analysis.

1 Introduction

Explanatory data analysis is the ...
All pictures in this paper are the built-in DataFrame visualizations that are created with Roassal2 and can be reproduced by following the steps in section

Paper structure In the first section I provide a brief introduction into the explanatory data analysis. What is it good for? How to do it right? I will also provide you with all the basic knowledge about statistics and data analysis, such as statistical variable, types of variables etc. In the second section I briefly describe the DataFrame - new data structure for data analysis. In the following section I will give a step-by-step example of how to perform EDA on the well-known Iris data set.

2 Exploratory Data Analysis

Everything that doesn't include fitting model to a data is an exploratory data analysis.

Here are the main reasons we use EDA:

- detection of mistakes
- checking of assumptions
- preliminary selection of appropriate models
- determining relationships among the explanatory variables
- assessing the direction and rough size of relationships between explanatory and outcome variables

3 DataFrame class

...

4 Exploring the Iris Data Set

Free form, variable number of sections, technical details.
But in general do not mix solution and discussions/possible variation let that for discussion
We can start by loading iris data set from a CSV file.

```
data := DataFrame fromCsv: '/path/to/iris.csv'.
```

DataFrame comes with a built-in collection of data sets that are widely used as examples for data analysis and machine learning problems. Iris is among them, so an alternative way of loading it would be simply

```
data := DataFrame loadIris.
```

Now let's take a look at the first and the last 5 entries in our table. These slices are called *head* and *tail* of a data frame.

```
data head.
data tail.
```

4.1 Univariate non-graphical EDA

To access a single variable we ask a data frame for a specific column, using its name or integer index

```
var := data column: #sepal_width. "Accessing column by its name"
var := data columnAt: 1. "Getting column by index"
```

What we can do with a column depends on a type of statistical variable it represents. If the variable is categorical, we can ..., if its quantitative, we can look at the ... statistics.

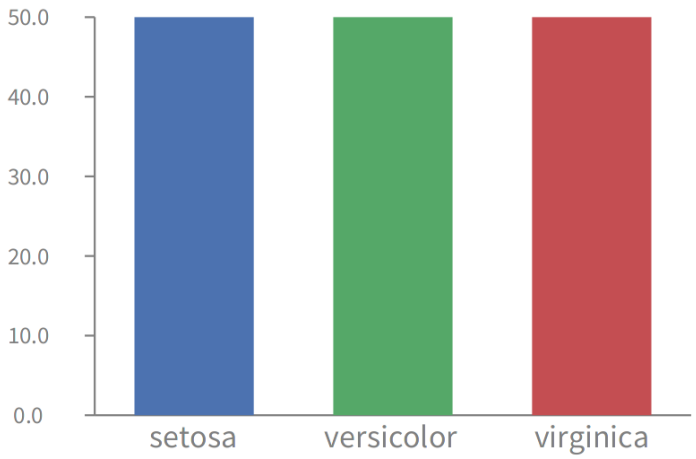
var min.
var max.
var range.
var average. "a.k.a. mean"
var median.
var mode.
var stdev.

4.2 Univariate graphical EDA

4.2.1 Categorical

Histogram is the only graphical technique that can be used for a categorical variable.

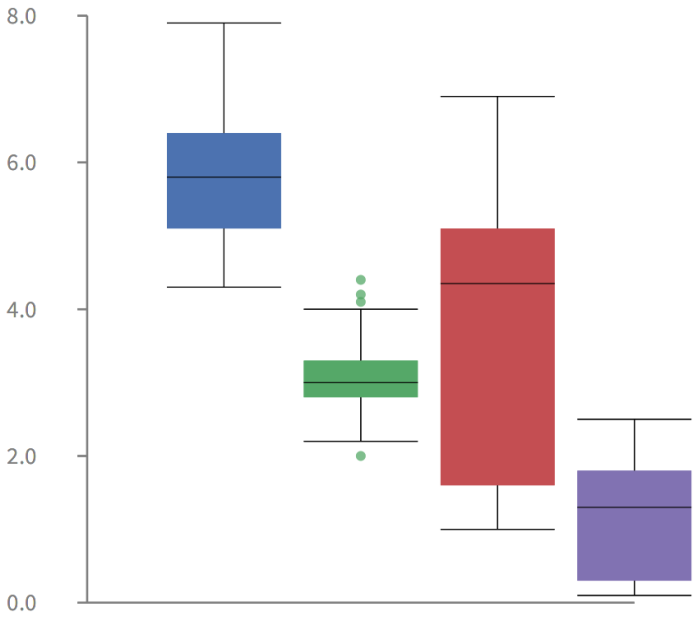
var := data column: #species.
var barplot.



4.3 Multivariate non-graphical EDA

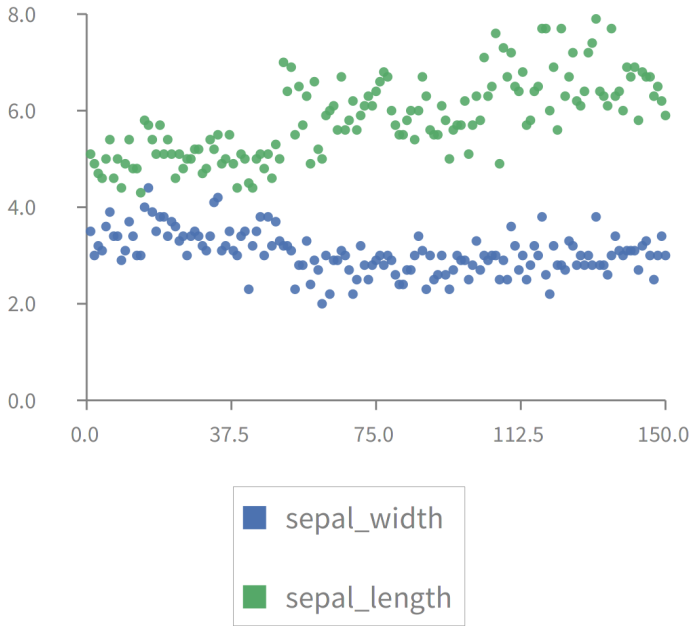
...

4.4 Multivariate graphical EDA



Let's look at the scatterplot of two statistical variables representing the width and length of a sepal. To do that we ask our DataFrame to give us specific columns, in our case, `sepal_width` and `sepal_length`, then we ask these columns (the result will be another DataFrame) to visualize themselves.

```
vars := data columns: #(sepal_width sepal_length).
vars scatterplot.
```



5 Discussion

Discussion of actual solution *vs.* initial constraints from **??**. Explain the space of the solution, why we made it this way.
Evaluation of the solution. How does the solution meet the criteria? Where does it succeed or fails...

6 Related Works

Other solutions in the domain, and a real comparison of our contribution with solutions from other people.

7 Conclusion

In this paper, we looked at problem P with this context and these constraints. We proposed solution S. It has such good points and such not so good ones. Now we could do this or that.

Acknowledgements

This work was supported by Ministry of Higher Education and Research, Nord-Pas de Calais Regional Council, FEDER through the 'Contrat de Projets Etat Region (CPER)

2007-2013', the Cutter ANR project, ANR-10-BLAN-0219 and the MEALS Marie Curie Actions program FP7-PEOPLE-2011- IRSES MEALS (no. 295261).