

[Open in app](#)[Get started](#)

Published in Towards Data Science

You have **2** free member-only stories left this month.

[Sign up for Medium and get an extra one](#)



Darío Weitz

[Follow](#)Jul 27, 2020 · 9 min read ★ · [Listen](#)

Save



Parallel Coordinates Plots

Why & How, Storytelling with Parallels





Open in app

Get started



[Open in app](#)[Get started](#)

AKA: Parallel Coordinates, Parallel Coordinate Charts, Parallel Plots, Profile Plots

WHY: A Parallel Coordinates Plot (PCP) is a visualization technique used to analyze **multivariate numerical data**. It allows data analysts to **compare** many quantitative variables together looking for **patterns** and **relationships** between them. They are appropriate for comparing multiple numerical variables simultaneously when those variables have different magnitudes (different scales) and different units of measurement. The idea is to find patterns, similarities, clusters, and positive, negative, or no particular relationships in multidimensional datasets.

The first PCP was published in 1885 (#1), but its popularity came a century later through the work of Inselberg (#2). They are widely used, particularly in academic papers, to overcome the challenges associated with the visualization of high-dimensional data. The n-dimensional capabilities of the PCP enable complex relationships to be plotted with simplicity (#3).

HOW: This representation does not show the classic Cartesian coordinate plane, but each numerical variable is given its own axis. As the following figure shows, all axes are placed **parallel**, vertical, and equally spaced. Every data element of the dataset is represented through **connected line segments**, derived from a connected set of points, one on each axis. We finally get a set of lines, each of which is a multi-axis representation of every data record. In general, lots of parallel lines indicate a positive relationship whilst lots of crossing lines (X shapes) indicate a negative association.





Open in app

Get started



63

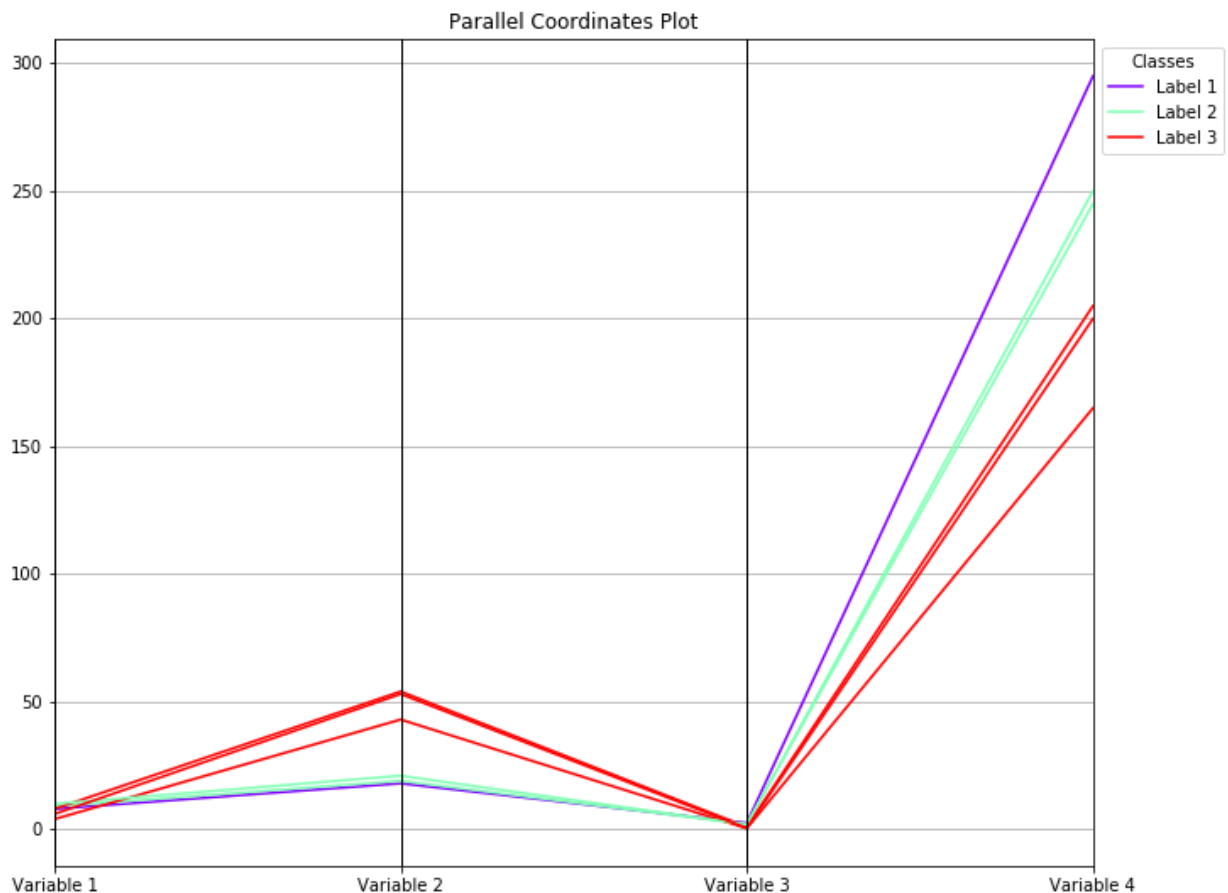


Fig. 1: Schematic diagram of a parallel coordinates plot. The figure was developed with Matplotlib.

Our example dataset has four numerical variables and six records related to three different classes. So we get four parallel axes and six lines (Fig. 1). Each line is derived from four connected points, one on each axis. Each class is represented by a particular colour. We are looking for patterns or relationships between the four numerical variables and the different classes. Clearly, we have a serious problem: the different orders of magnitude between variable 4 and variables 1 and 3 do not allow us to discover these possible patterns.

To solve this problem we must proceed with a **data normalization task** prior to plot the data. Remember that raw data may have not only different orders of magnitudes but also different units of measurement. Normalization (scaling) transforms raw data into a new scale that allows us to compare values of initially very different magnitudes.





Open in app

Get started

in between are transformed accordingly. For example, after scaling with this procedure, you can compare with a PCP two numerical variables with one of them initially in the range 0–10, and the other between 100 and 10000. Other scaling techniques use the mean and standard deviation, the median, or another statistic to transform the original numerical values into a common scale.

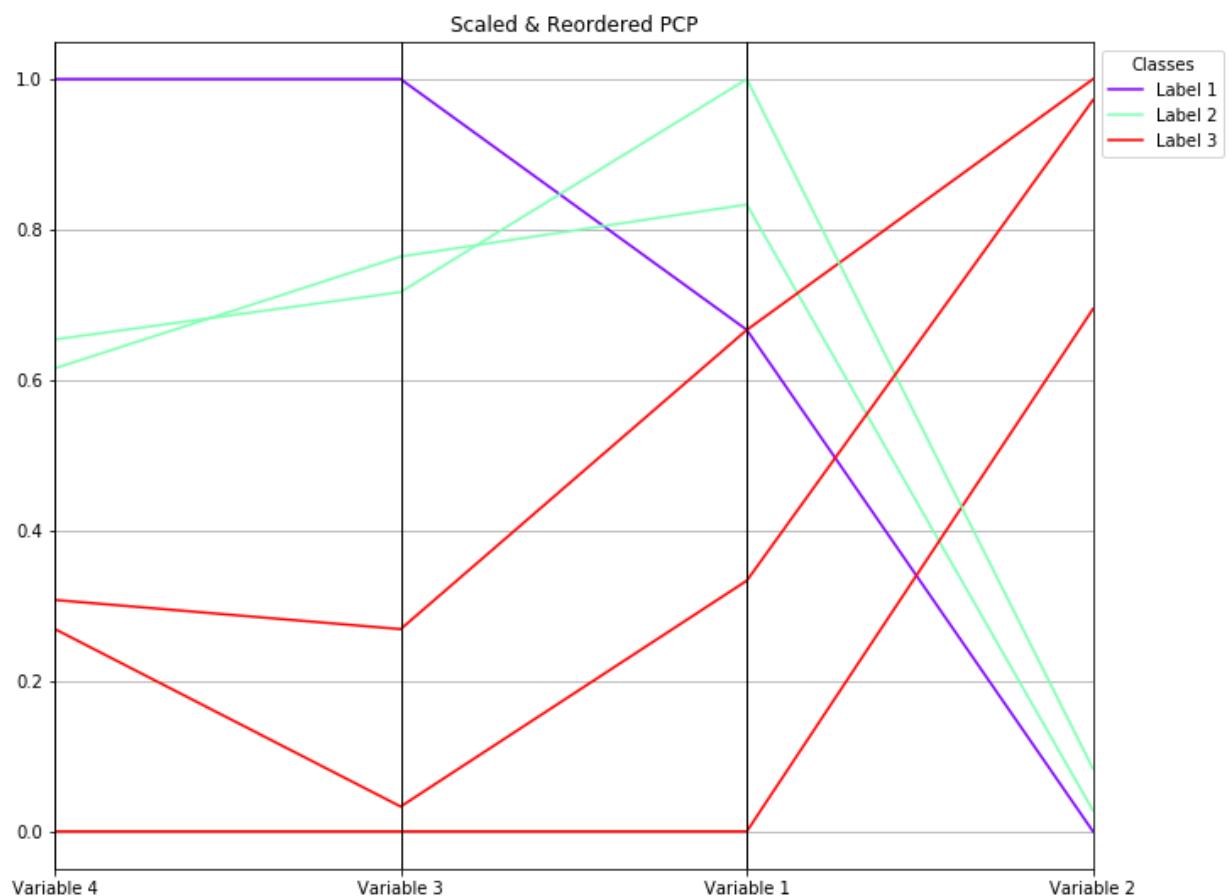


Fig. 2: Scaled and reordered parallel coordinates plot. The figure was developed with Matplotlib.

Fig. 2 uses the same data as Fig. 1 but every axis is scaled between 0 and 1. We played with some reordering (described below) until we notice that Label 1 and Label 2 share some kind of positive relationship while Label 3 shows an opposite pattern.

Besides scaling, three other techniques are usually employed in a PCP in order to discover patterns or relationships across variables: colouring; reordering; brushing.



[Open in app](#)[Get started](#)

Reordering is changing the order of the vertical axes. The reason behind it is that relationships between adjacent variables are more easily visualized than between non-adjacent ones. Then, you may not discover some kind of pattern or relationship simply because the variables are not located adjacent to each other. Besides, you can reduce the clutter of your PCP by changing the order of some axes and minimizing the number of crosses between them. Modern visualization tools allow us to drag the axes along the plot to facilitate the reordering. A clever analyst **always experiments with reordering** until he enhances the readability of the plot and gets as much information as possible from the display.

Brushing is a technique where individual data points are selected by an application of a *brush* for highlighting subsets of the data. Selected lines are emphasized while others are faded. A brush is like a filter that reduces the number of lines, minimizing clutter and revealing patterns in the dataset. Brushing is mandatory on PCP to avoid overplotting and occlusion during the analysis of huge datasets.

The following figure clearly indicates the utility of the brushing technique: top image (a) shows the PCP over-plotted with data; bottom image (b) shows the same PCP with a brush applied to it. The original dataset corresponds to a telecommunications industry recorded by a startup specialized in call center data collection and analysis (#3). The dataset contains five million records related to customer interactions (phone calls) with call centers. There are nine numerical variables, each with its corresponding axe in the PCP. The brushing allows the company to identify callers who couldn't talk long enough to solve any problem they might have and provided the telecommunications experts with previously unobserved patterns. It is evident that the pattern could not be observed in the original over-plotted PCP.





Open in app

Get started

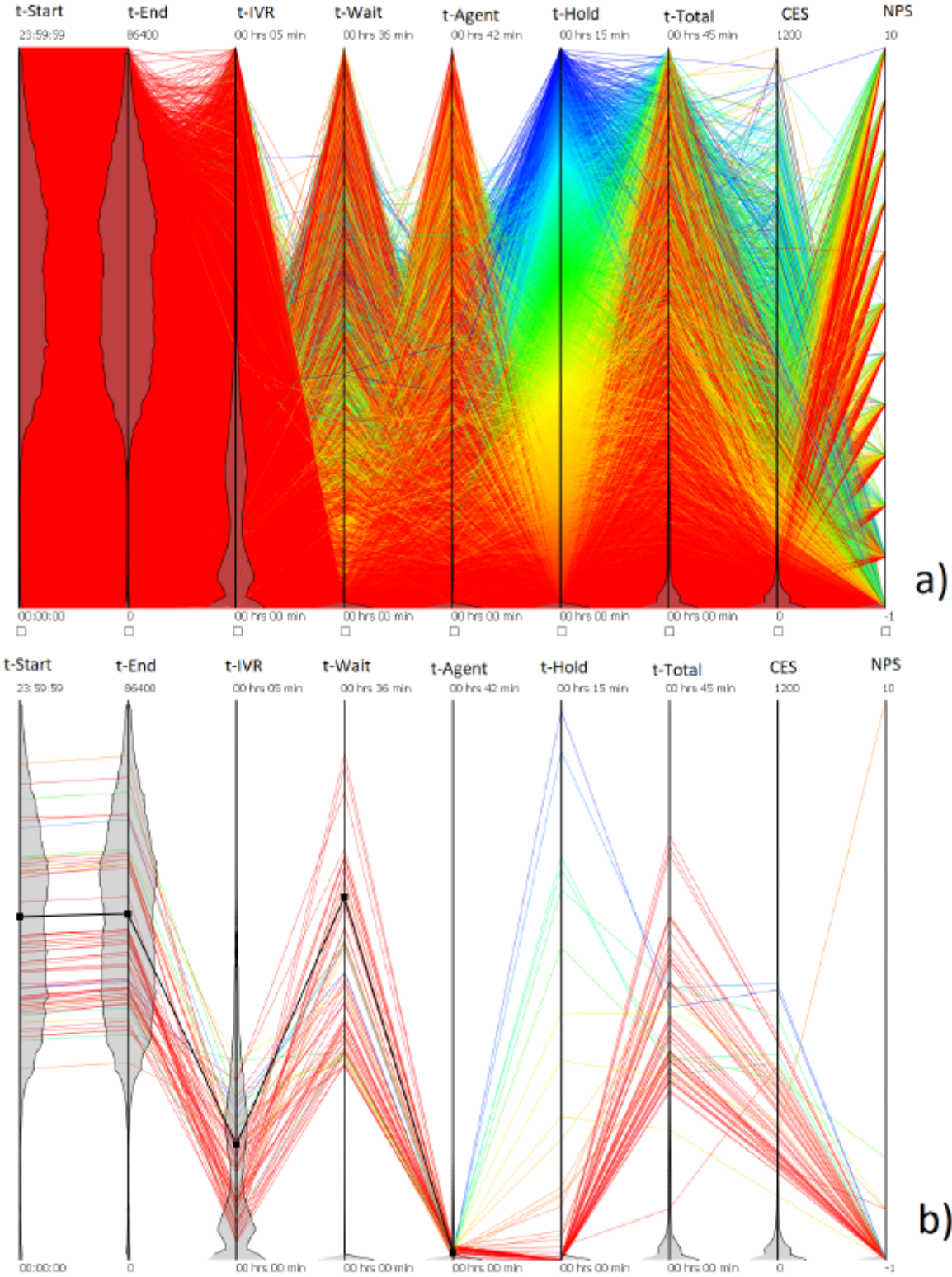


Fig. 3: a) PCP before brushing; b) PCP after brushing, Source (#3).

Warnings

[Open in app](#)[Get started](#)

Try not to show many numerical variables at the same time. More than twelve axes and their corresponding lines (even with brushing) in the screen might confuse the audience;

One downside to this display is that after scaling we lose the original values of every variable.

PCPs are not suitable for **categorical data**. Remember that a categorical variable, also called qualitative variable, is one that usually takes a limited number of values of mutually exclusive categories or groups. These values can be numerical but do not represent quantities but mutually exclusive groups (i.e. marital status: 1- Never Married; 2- Married; 3- Divorced; 4- Widowed). You must use **Parallel Sets Plots** with categorical variables;

A radar chart (spider chart, web chart) is another method for displaying multivariate data of three or more quantitative variables. The problem with radar charts is that they are based on non-common axes and circumferential gridlines which makes the task of comparison very difficult.

It is easy to mislead PCP with line plots, but they tell different stories. The ideas behind brushing and reordering are specific to PCP. Furthermore, time series analysis is not appropriate with PCP because there is no possibility of reordering.

In summary: relationships between adjacent axes can be easily seen, but not between nonadjacent axes. Hence, revealing data patterns often requires reordering axes and careful choice of a color map or a brushing filter. A positive (negative) relationship is suggested between two variables when lines between two parallel axes are somewhat parallel (X shape) to each other.

Storytelling with Parallel Coordinates Plots: chemical properties affecting the quality of wine

In an article previously published in Towards Data Science (“Bubble Charts, Why & How, Storytelling with Bubbles”, <https://towardsdatascience.com/bubble-charts-why-how-f96d2c86d167>) we analyzed data from a Kaggle Competition related to wine quality exploration and analysis of red and white variants of the Portuguese “Vinho





Open in app

Get started

and red wines. The purpose of the project was to evaluate which of the following chemical or physical properties influence the quality of the wines: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulfates, and alcoholic grade. The set of wines was evaluated by three experts who provided quality scores between 0 (Bad) and 10 (Excellent) for each wine. Table 1 shows the first five records of the dataset.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	color
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	R
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5	R
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5	R
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6	R
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	R

Table 1: First five records of the wine dataset

At the end of the article, we claimed that alcoholic grade and residual sugar level were the two most important parameters to differentiate between a low-quality wine from a high-quality one. Now it remains to be seen if any of the other chemical parameters play a preponderant role in determining the quality of the final product.

To answer that question we decided to use a PCP because the remaining eight chemical properties were recorded as quantitative variables and we wanted to analyze them simultaneously.

We developed the following procedure:

****** First install Anaconda, a cross-platform Python Distribution for tasks like Python computing and data analysis. Then import the following libraries: Pandas, Numpy, Matplotlib, Scipy, and Scikit-learn. Use the `read_csv()` function to read and parse the file. Check for missing records and look for the shape of the dataset: there are 4898 records and 13 columns. Ten columns correspond to chemical properties (fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, pH, sulfates, and alcoholic grade). One corresponds to a physical property





Open in app

Get started

** Next eliminate anomalous values (outliers) because they could have a significant effect during the normalization process. Use `scipy.stats.zscore()` and experiment with different threshold values;

** Split up the dataset into high-quality wines (quality scores 7, 8, and 9) and low-quality wines (quality scores 3, 4, and 5);

** Proceed with the scaling stage utilizing the scikit-learn method *MinMaxScaler*. Remember that *MinMaxScaler* rescales the data from its original value so all the new values are within the range 0–1 (# 4);

** Use the *parallel_coordinates* plotting function, Pandas built-in plotting function for creating a parallel coordinates chart using Matplotlib. Required arguments are *frame*, *class_column*, and *colormap*. Frame includes the following nine columns (quality scores + eight chemical properties): quality, fixed acidity, volatile acidity, citric acid, chlorides, free sulfur dioxide, total sulfur dioxide, pH, and sulfates. Quality is the *class_column* (Column name containing class names). Select two different colormaps, one for high-quality wines and the other for the low-quality wines plot;

** Experiment with reordering.

We obtained Fig. 4 for high-quality wines and Fig. 5 for low-quality wines. High-quality wines show lower values for chlorides and free sulfur dioxide, and higher levels for fixed acidity and total sulfur dioxide. Low-quality wines show lower values for free sulfur dioxide, and higher levels for chlorides and volatile acidity.



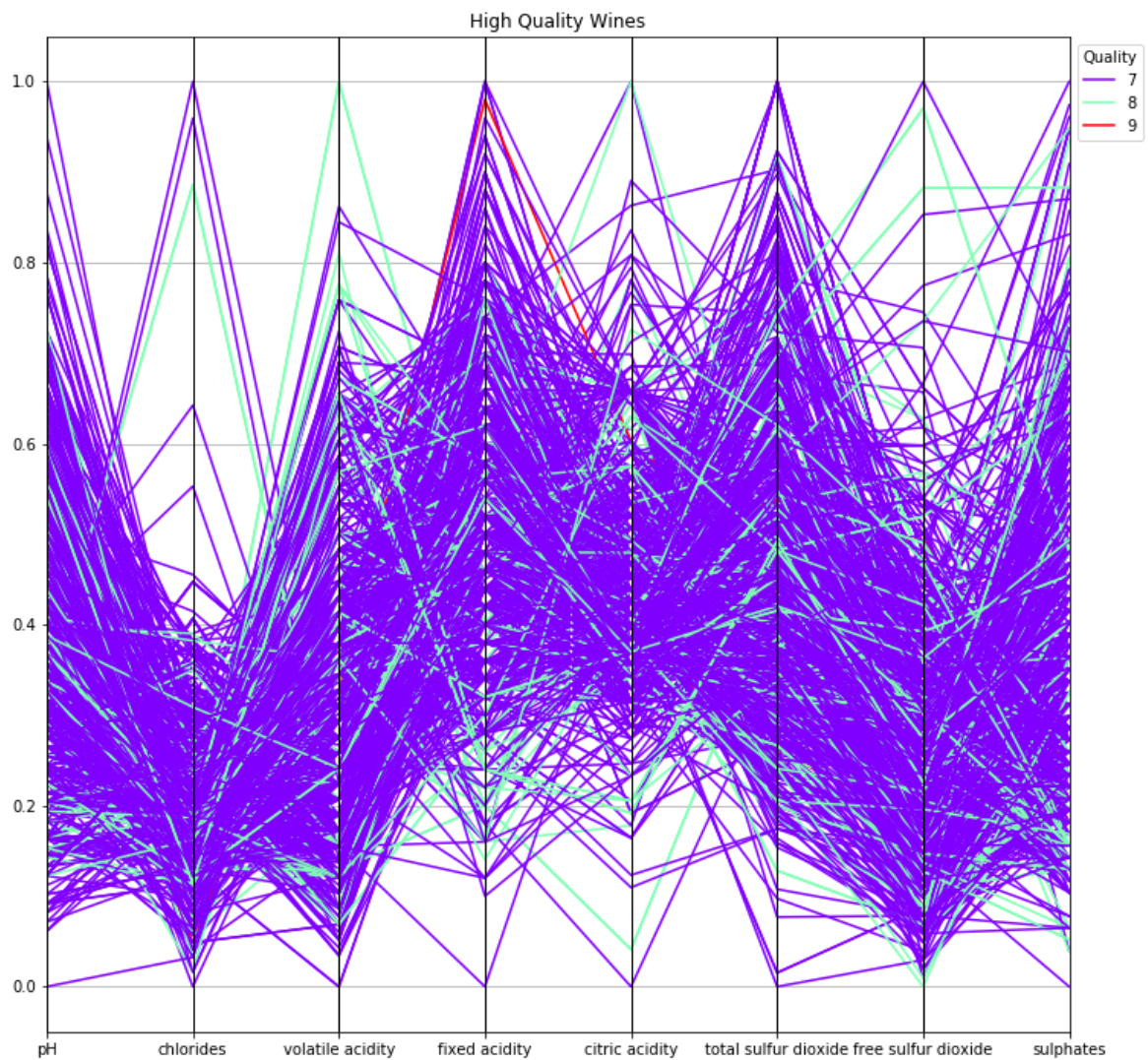
[Open in app](#)[Get started](#)

Fig. 4: PCP for high-quality wines. The figure was developed with Matplotlib.





Open in app

Get started

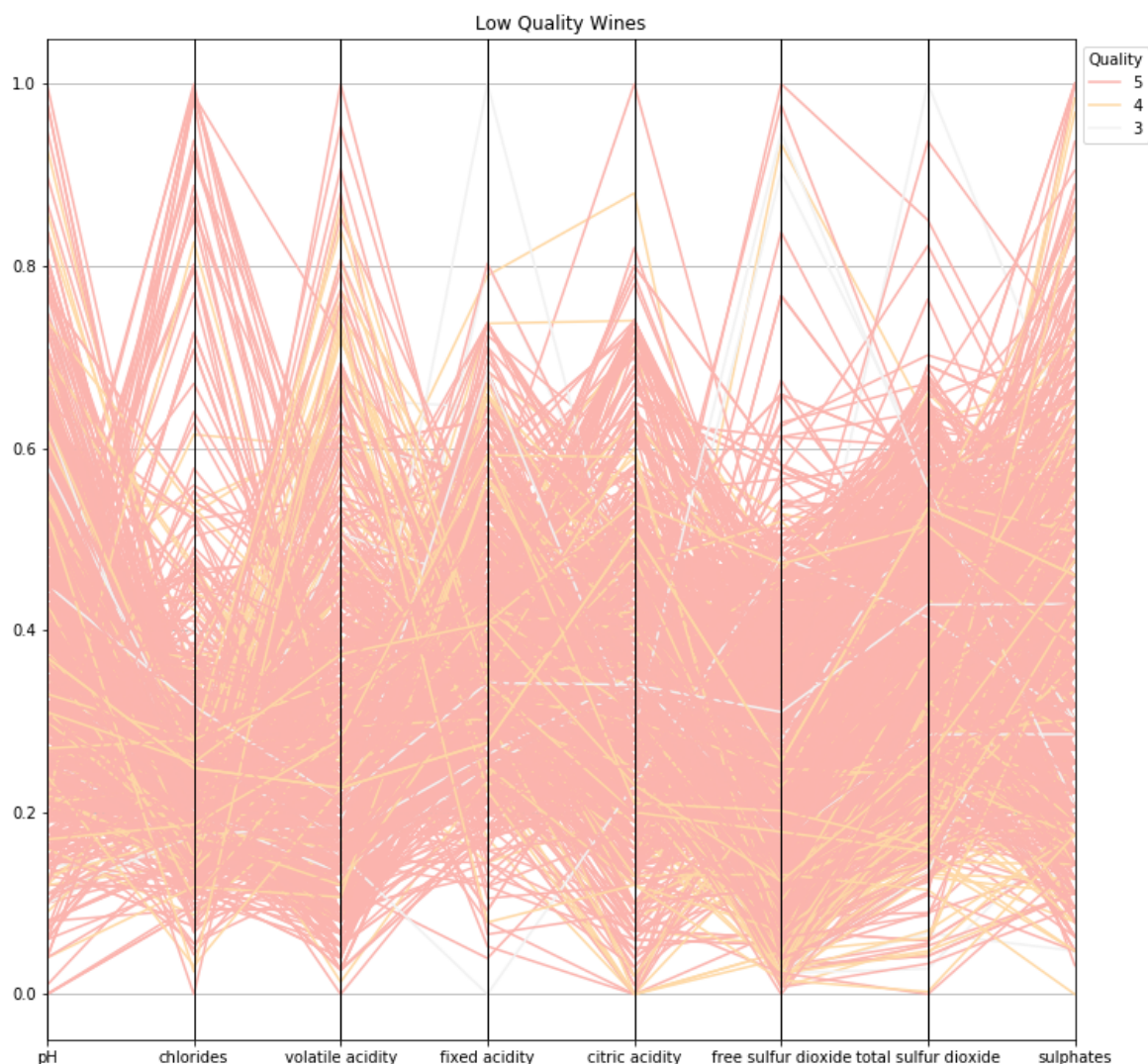


Fig. 5: PCP for low-quality wines. The figure was developed with Matplotlib.

The wine industry uses **sulfur dioxide** (SO₂) for its antioxidant and antimicrobial properties and to prevent color changes, especially in white wines. Total sulfur dioxide is a measure of both the free and bound forms of SO₂. Sulfur dioxide also improves the taste and retains the wine's fruity flavors and freshness of aroma (#5). The use of SO₂ has become a controversial issue due to the gradual documentation of adverse reactions in consumers, who may have mild allergies to its presence (therefore, the indication of its use on the label has become part of some wine regulations).





Open in app

Get started

compounds in types of acid found in wine, showing an aroma, rather than found on the palate. Excessive amounts of volatile acidity cause undesirable sensory effects (#5).

Several chemical properties are responsible for the organoleptic characteristics of wines. We found that sulfur dioxide, chlorides, and volatile acidity play an important role in determining the quality of the final product. As those properties can be more or less quantified, PCP can help producers to develop high-quality wines for a broader number of consumers.

If you find this article of interest, please read my previous:

Clustered & Overlapped Bar Charts, Why & How

<https://towardsdatascience.com/clustered-overlapped-bar-charts-94f1db93778e>

Stacked Bar Graphs, Why & How, Storytelling & Warnings

<https://towardsdatascience.com/stacked-bar-graphs-why-how-f1b68a7454b7>

References

#1: M d'Ocagne, "Coordonnees paralleles & axiales: methode de transformation geometrique et procede nouveau de calcul graphique deduits de la consideration des coordonnees paralleles". Gauthier-Villars, 1885.

#2: A. Inselberg, "N-dimensional coordinates," Picture Data Description & Management, p. 136, 1980.

#3: R. Roberts et al, "Smart Brushing for Parallel Coordinates", Journal Of Latex Class Files, vol. 14, no. 8, August 2015.

#4: J. Brownlee, "How to Use StandardScaler and MinMaxScaler Transforms in Python", <https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/>

#5: <https://www.decanter.com/>



[Open in app](#)[Get started](#)

More from Towards Data Science

[Follow](#)

Your home for data science. A Medium publication sharing concepts, ideas and codes.

Manoj Kukreja · Jul 27, 2020 ★

Manoj

Database Migration using AWS Data Migration Service (DMS) — A few lessons...

AWS 6 min read

Database Migration using AWS Data Migration Service (DMS)—A few lessons learnt along the way

Ramya Vidiyala · Jul 27, 2020 ★

Ramya

Feature Engineering in Machine Learning

Machine Learning 6 min read

Feature Engineering in Machine Learning

Chanin Nantasenamat · Jul 27, 2020 ★

Chanin

The Data Science Process

Data Science 7 min read

The Data Science Process

Robert Wood · Jul 27, 2020 ★

Robert

Visualizing Multiple Regression in 3D

Data Science 6 min read

Visualizing Multiple Regression in 3D

Akash Kaul · Jul 27, 2020 ★

Akash

Designing a 3D Healthcare Network Graph


Graph 6 min read

Designing a 3D Healthcare Network Graph

[Read more from Towards Data Science](#)



Recommended from Medium

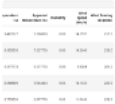
 Dorian Lazar in Analytics Vidhya


Publishing your first dataset on Kaggle



 Puneet

Performing Analysis of Meteorological Data



 Sune Dupont in Destination AARhus-TechBlog

Listen to your Pipes or How Acoustic Measurements are used to Locate Water Leaks



 Jonah Flateman

From Records Management to Data Science

 Wouter Trappers in Plumbers Of Data Science


Data lineage using a Graph Database



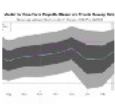
 Maximilian Joas


How Data Science Can Make Hotels More Profitable



 Graham Giller in Adventures in Data Science

Are Private Survey Estimates of Macroeconomic Data Accurate?



 Josh Benamram in Databand.ai

3 Steps to Advanced Alerting on Airflow with Databand





Open in app

Get started

