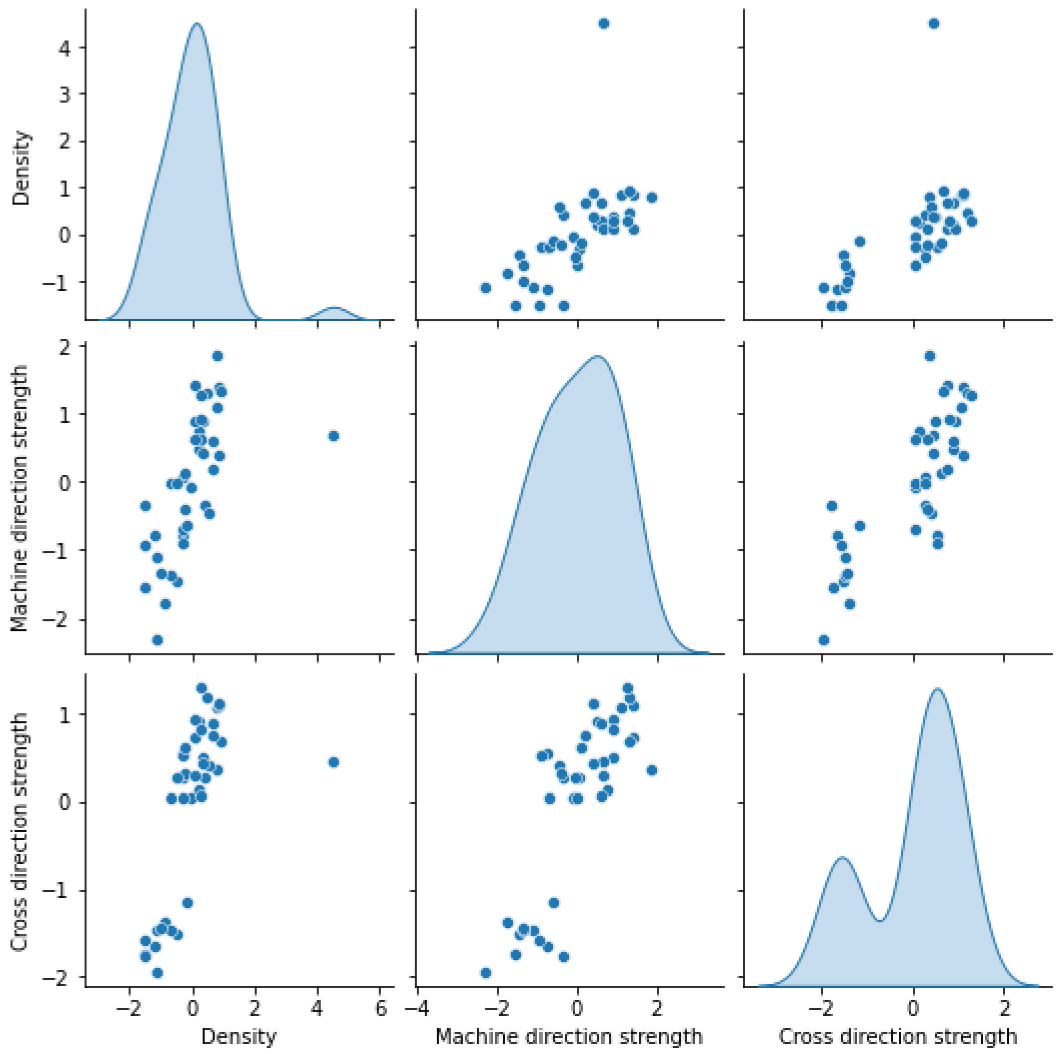
# Multivariate Analysis
# Graphical Plots

# Scatterplot Matrix (SPLOM)

- Scatterplot Matrix plots all possible combinations of two or more numeric variables against one another.
- The plots are arranged in rows and columns, with the same number of rows and columns as there are variables. The point of the plot is simple. When you have many variables to plot against each other in scatterplots, it is logical to arrange the plots in rows and columns using a common vertical scale for all plots within a row (and a common horizontal scale within columns). All complete x-y pairs within each plot are used; that is pairwise deletion is used for missing data.

| | Density | Machine direction strength | Cross direction strength |
|---|---|---|---|
| **0** | 0.801 | 121.41 | 70.42 |
| **1** | 0.824 | 127.70 | 72.47 |
| **2** | 0.841 | 129.20 | 78.20 |
| **3** | 0.816 | 131.80 | 74.89 |
| **4** | 0.840 | 135.10 | 71.21 |

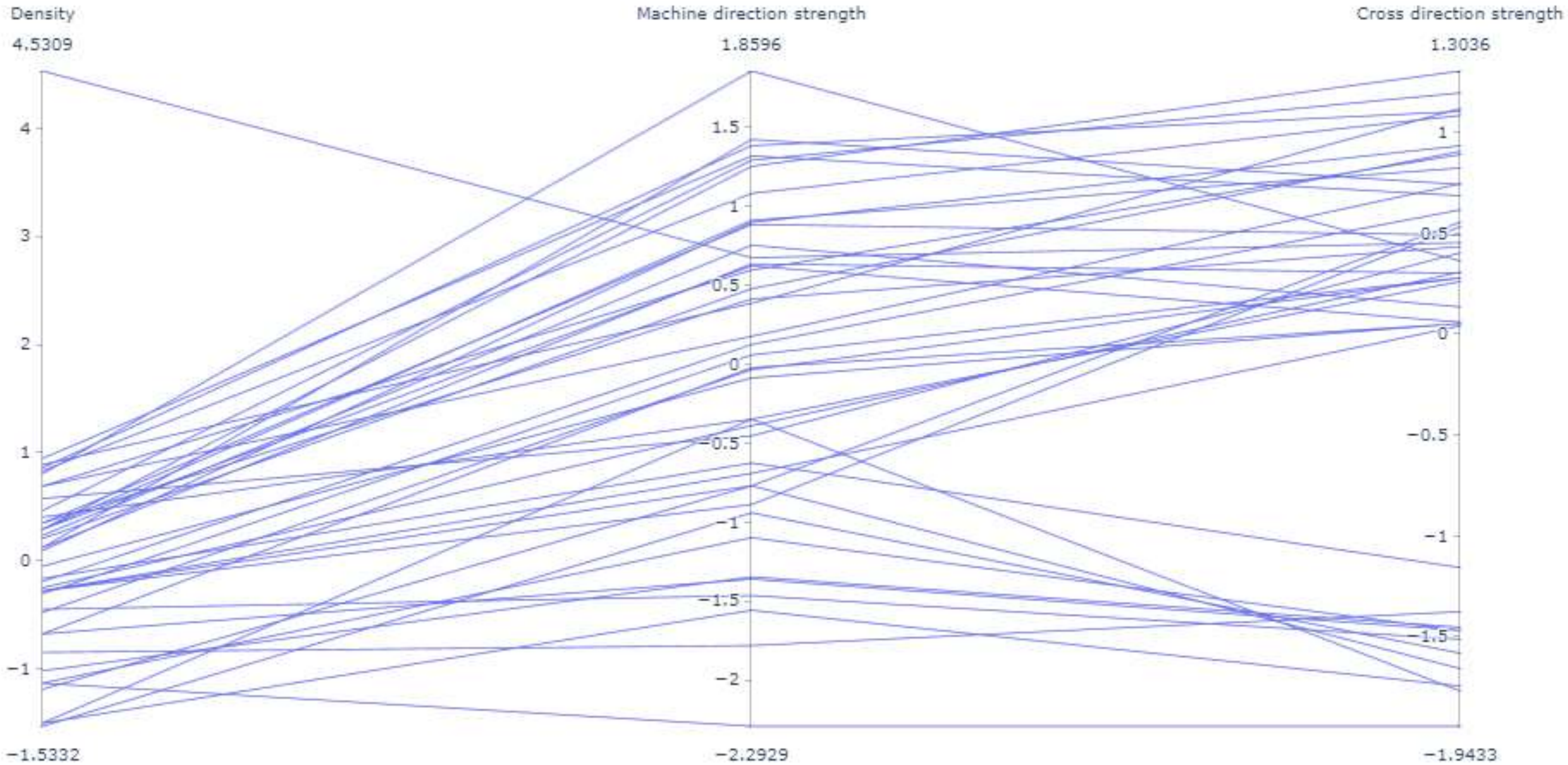
```
from scipy import stats
dfz = stats.zscore(df)

import seaborn
seaborn.pairplot(dfz,
kind='scatter',diag_kind="kde",palette="deep")
```



# Parallel Coordinate

- Cartesian coordinates are computed on perpendicular axes. This works fine for two- and three-dimensional plots. Higher dimensions, however, would be difficult to visualize. Inselberg (1985) proposed making coordinate axes parallel for these higher dimensional plots.
- A parallel coordinate display draws an axis for each variable and positions them side by side so they are parallel. The same scale is used for each axis. The values for a case are plotted on the axes and connected with a line segment.
- The procedure for **Parallel** is similar to that for **Fourier**. Indeed, parallel and Fourier coordinates are alternative representations of the same data. The advantage of Fourier coordinates is that variables are reduced to a smaller number of features in the plot. The advantage of parallel coordinates is that plot shows the variables themselves, which facilitates interpretation.

```
import plotly.express as px
fig = px.parallel_coordinates(dfz)
fig.show()
```



# Andrews Fourier Plot Dialog Box

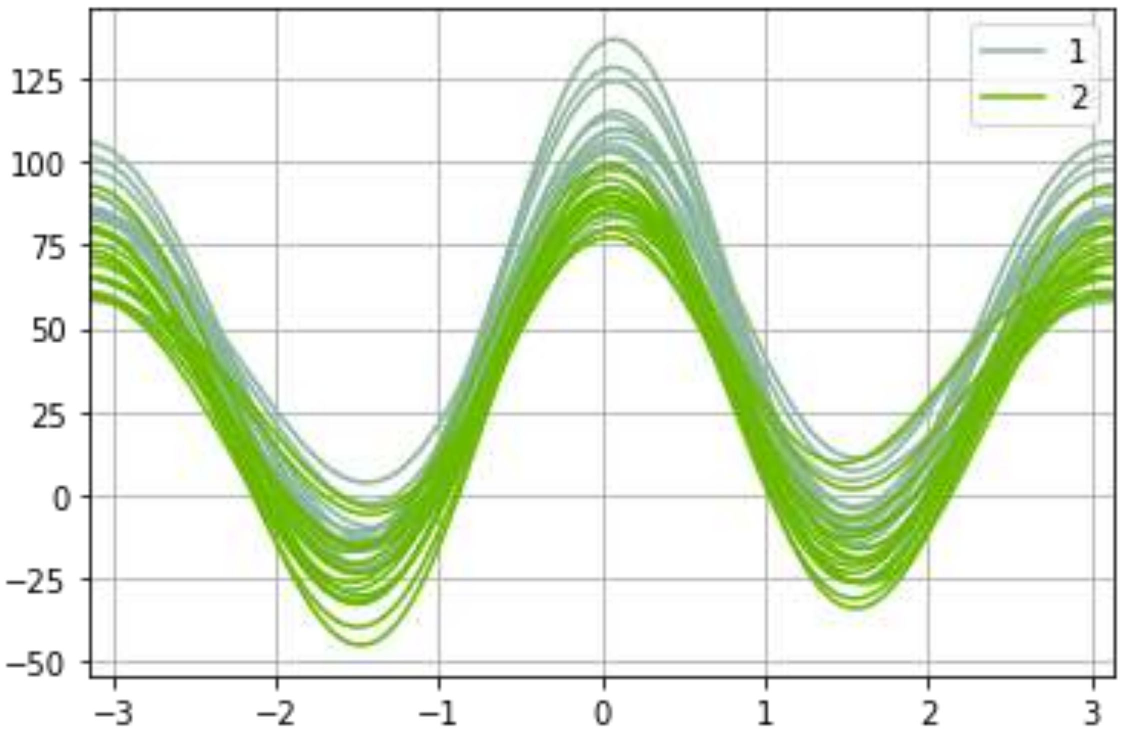
A particularly powerful method for identifying clusters of cases in multivariate data is to plot their Fourier components. Andrews (1972) developed these plots. Fourier functions have the following form:

$$f(t) = \frac{y_1}{\sqrt{2}} + y_2 \sin(t) + y_3 \cos(t) + y_4 \sin(2t) + y_5 \cos(2t)\ldots$$

where y is a p-dimensional variate and t varies from -3.14 to 3.14 (1/4 radians on either side of 0). The result of this transformation is a set of wave forms made up of sine and cosine components for each selected variable.

Each wave form corresponds to one case in the data file. Cases that have similar values across all variables will have overlapping wave forms in the plot. Cases with different patterns of variation will have contrasting wave forms

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
dfb =
pd.read_csv("C:/Users/sande/Desktop/Python/DAdata.csv")
x = pd.plotting.andrews_curves(dfb,'visit')
x.plot()
plt.show()
```



# Radar Chart

- **Radar** are profile icons in polar coordinates;
- The distance of each point from the center of the icon shows the value of the corresponding variable.
- Separate icons are drawn for each case.

```
import plotly.express as px
import pandas as pd
df = pd.DataFrame(dict(
    r=[1, 5, 2, 2, 3],
    theta=['processing cost','mechanical properties','chemical stability’, 'thermal stability', 'device integration']))
fig = px.line_polar(df, r='r', theta='theta', line_close=True)
fig.show()
```