Principia Pictura

By Ismaël Chang Ghalimi

Principia Pictura received major contributions from Christian Kaas, Florian Ribon, and François Beaufils.

Many thanks to them.

Work on *Principia Pictura* was funded by Sutoiku, Inc.

Many thanks to their customers and investors.

Questions and comments should be sent to ismael@stoic.com.

chart

: visualization of data dimensions and measures through geometrical marks such as dots, lines, or bars aligned against axes for measurement and comparison purposes.

Principia Pictura is a unified grammar of charts. Its purpose is to facilitate the visualization and analysis of structured datasets by suggesting the most appropriate visuals and automating the binding of dataset variables onto chart axes. *Principia Pictura* is built on top of the *Principia Data* unified typology of statistical variables. *Principia Pictura* draws its inspiration from the fields of computer science and statistics, but was not vetted by any formal peer review process.

Table of Contents

Table of Contents 3

Audience 7

Background 8

Charts, Maps, Visuals, and Infographics 9

Cognitive Context 10

Cultural Context 11

Structure of a Chart 12

Primary Charts 13
- Single Dimension 14
- Single Measure 16
- Summary of Primary Charts 17
- Special Primary Charts 18

Secondary Charts 19
- One Dimension and One Measure 20
- Two Dimensions 24
- Two Measures 25
- Tabular Secondary Charts 27
- Absolute Secondary Charts 30
- Differential Secondary Charts 32

Tertiary Charts 33
- Three or More Dimensions 34
- Two Dimensions and One Measure 35
  - Nominal Measure 35
  - Ordinal Measure 35
  - Intensive Measure 36
  - Extensive Measure 36
  - Absolute Measure 37
- One Dimension and Two Measures 38
  - Two Discrete Measures 38
  - One Discrete and One Continuous 38

Two Intensive Measures 39
One Intensive and One Extensive 39
Two Extensive Measures 40
Three or More Measures 41

Quaternary Charts **42**
Two Discrete Measures 42
One Discrete and One Continuous 42
Two Continuous Measures 43

Chart Axes **44**
Axis Relation 45
Axis Typology 46
Signable Axis 47
Required Axis 48
Axis Multiplicity 49
Cohorts 50
Details 51
Columnar and Tabular Axes 52
Latitudinal, Longitudinal, and Altitudinal Axes 53
Differential Axes 54
Axis Scale 55
Axis Drill-Down 56
Axis Brushing 58

Chart Marks **59**
Mark Types 60
Rectangles 61
Areas 62
Lines 63
Paths 64
Arcs 65
Symbols 66
Images 67
Mark Colors 68
Boolean Color Palette 68
Nominal Color Palette 68
Ordinal Color Palette 69
Diverging Color Palette 69

Gradient Color Spectrum 69
Mark Drill-Down 70
Mark Brushing 71

Variable Binding 72

Chart Classification 73
Cartesian Charts 74
Bar Chart 75
Level Chart 76
Area Chart 77
Line Chart 78
Block Chart 79
Segment Chart 80
Ribbon Chart 81
Range Chart 82
Circle Chart 83
Scatter Plot 84
Tabular Charts 85
Dot Chart 86
Bit Chart 87
Castle Chart 88
Band Chart 89
Train Chart 90
Tape Chart 91
Gantt Chart 92
Heatmap Chart 93
Table Chart 94
Geographics Maps 95
Choropleth Map 96
Location Map 97
Other Visuals 98
Calendar 99
Arc Plot 100
Parallel Coordinates 101
Contour Plot 102
Surface Plot 103

Chart Recommendations 104

Visual Transformations **106**
Chart Orientation 107
Chart Tabulation 108
Chart Polarization 109
Chart Composition 112
Chart Juxtaposition 113
Chart Superimposition 114
Chart Tiling 115
Mark Subdivision 116
Mosaic Subdivision 117
Treemap Subdivision 118
Partition Subdivision 119

Visual Enhancements **120**
Groups 121
Clusters 122
Highlights 123
Differences 124
Growths 125
Trends 126
References 127
Labels 128
Summaries 129
Texts 129
Breaks 131
Connectors 132
Paddings 133

Missing Data **134**

Principia Pictura Framework **135**

Conventional Statistical Plots **137**

Audience

This document is aimed at anyone interested in learning more about the fundamental nature and structure of charts, for the purpose of improving one's understanding of common visualization techniques and one's ability to analyze and visualize structured datasets.

Formal training in the disciplines of computer science or statistics are not required *a priori*, and could in fact prevent a careless reader from keeping an open mind about the subject at hand. Nevertheless, the reader is strongly encouraged to refer to other reference materials whenever some notions deserve deeper investigations.

Principia Pictura is built on top of the *Principia Data* unified typology of statistical variables. Therefore, thorough reading and understanding of the *Principia Data* framework are mandatory prerequisites before attempting to learn about the concepts introduced by *Principia Pictura*.

Principia Pictura makes the assumption that charts defined in accordance with its grammar must be produced by some kind of **chart rendering engine**. The rules, heuristics, and algorithms defined in this paper are aimed at being implemented by such an engine.

Background

The first semiology of graphics was authored by Jacques Bertin in 1967 with a book aptly titled Semiology of Graphics. Later on, a first grammar of graphics was proposed by Hadley Wickam in a seminal 2010 article titled A layered grammar of graphics[1]. It was quickly followed by the Vega Visualization Grammar developed by Jeffrey Heer. Unfortunately, none were built on top of a formal typology of statistical variables, which prevented them from reaching a sufficient level of genericity. *Principia Pictura* addresses this shortcoming by using the *Principia Data* unified typology of statistical variables as underlying foundation.

Principia Pictura was designed to address the following requirements in a coherent manner:

- Providing a formal definition of charts, maps, and other visuals;
- Enumerating and defining all basic charts;
- Defining visual transformations for producing advanced charts;
- Binding the variables of datasets onto the axes of charts;
- Defining compatibility rules between variable typologies and axis properties;
- Recommending the most appropriate chart for a given dataset;
- Producing conventional statistical plots with standard charts;
- Suggesting common user interactions for producing charts with software tools.

[1] Journal of Computational and Graphical Statistics, vol. 19, no. 1, pp. 3–28, 2010.

Charts, Maps, Visuals, and Infographics

A picture is worth a thousand words, but are all pictures worth the same?

Principia Pictura focuses on pictures that can be drawn by following a formal process, such as **charts**, geographic **maps**, and other structured **visuals**. In so doing, it ignores less structured visualizations usually referred to as **infographics**, which tend to favor cosmetic decorations at the expense of simplicity and legibility. In the remainder of this document, the terms *chart* and *visual* will be used interchangeably, even though the latter is more generic than the former.

Cognitive Context

In 1984, statisticians William Cleveland and Robert McGill published the seminal paper Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods[2]. In this paper, the authors study the cognitive processes people use to understand a chart and rank them, from the easiest to the most challenging:

1. Position along a common scale (bar chart, dot plot)
2. Position along nonaligned, identical scales (differential charts)
3. Length, direction, angle (pie chart)
4. Area (treemap)
5. Volume, curvature (3D bar chart, surface plot)
6. Shading, color saturation (heatmap, choropleth map)

For example, comparing positions along a common scale is much easier than comparing angles. Therefore, a *bar chart* should almost always be prefered to a *pie chart*. Leveraging this fundamental *hierarchy of cognitive processes*, simple heuristics can be implemented to rank multiple charts used to visualize the same dataset, putting the easiest to interpret at the top.

Principia Pictura is combining this hierarchy with the unified typology of statistical variables defined by *Principia Data* to automate the production of statistically correct charts for the effective visualization of arbitrary datasets.

[2] Journal of the American Statistical Association, Vol. 79, No. 387 (Sep., 1984), 531-554. [PDF]

Cultural Context

While the cognitive context introduced above is important, it must be balanced with the cultural context within which *Principia Pictura* is introduced. Some charts might be less effective than others from a cognitive standpoint, but their cultural familiarity might make them suitable or even desirable in certain situations.

For example, a *pie chart* (or *donut chart* preferably) would be acceptable when just 2 or 3 values must be visualized, their relative differences are significant, no value is so small in relation to the others that it could not be easily discernable, the real-estate available to display the chart is quite limited, and the use of a color palette could bring a desirable aesthetic enhancement to an otherwise uninspiring dashboard or report.

To make a long story short: if all the stars are aligned, get a donut; otherwise, stick to bars.

Structure of a Chart

As mentioned in preamble, a chart is defined as the visualization of data dimensions and measures through geometrical marks such as *dots*, *lines*, or *bars* aligned against axes for measurement and comparison purposes. Therefore, the three main components of a chart are:

- **Axes** — against which marks are aligned;
- **Marks** — through which data dimensions and measures are visualized;
- **Bindings** — with which data dimensions and measures are bound to axes and marks.

With such a simple definition, a casual reader might conclude that charts are simple entities. Such a conclusion would be tragically misleading though, for there is more than meets the eye. Charts might seem simple because they are pervasive: our visual landscape has been full of them for what might seem like forever. Nevertheless, this apparent familiarity should not be confused with some real intimacy, for charts remain poorly understood by most, even within the growing communities of statisticians or data analysts.

With such a backdrop, the primary purpose of *Principia Pictura* is to foster this intimacy.

In what follows, *dimension* and *independent variable* are synonyms.

Similarly, *measure* and *dependent variable* are synonyms.

Primary Charts

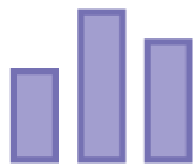
Principia Pictura aims at answering a simple question:

> *"How should this dataset be visualized?"*

One way to answer the question is to consider some sample datasets, from the simplest to the most complex, and to answer it for each and every one of them. To get started, one can consider univariate datasets, which are datasets made of a single variable or column of data. For these, two cases need to be considered: single independent variable and single dependent variable. And for each of them, *Principia Pictura* offers a set of **primary charts**.

Single Dimension

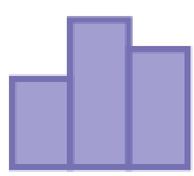
By convention, when a dataset made of a single independent variable and a single dependent variable is visualized, the former is usually depicted against the horizontal axis, while the latter is depicted against the vertical axis. Keeping with this convention, the single independent variable should be depicted against the horizontal axis. By definition, an independent variable is always discrete, and the simplest discrete variable is a *nominal* one. In that case, the simplest chart that can be drawn is a **frequency chart**.



Frequency Chart

A frequency chart is a special case of **bar chart**, for which each bar represents a possible value of the discrete variable, and the height of the bar is proportional to the frequency of the value within the variable. This chart can be produced by computing a pivot with the variable as sole dimension, and a *COUNT* as aggregation. In other words, the bar associated to a value of the variable displays the number of times the value can be found within the variable's list of values.

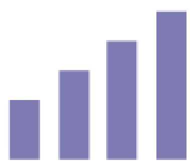
A similar chart can be produced when the variable becomes *incremental*. In that case, one would want to depict some form of continuity on the horizontal axis, and this could be achieved by removing the horizontal padding displayed between vertical bars on the frequency chart.



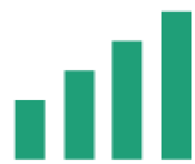
Histogram

This chart is called a **histogram**, and displays the exact same information as the frequency chart introduced above. Nevertheless, this simple example illustrates how minute visualization details like padding can play a critical role in conveying major differences regarding typologies.

By convention, the frequency and histogram charts are displayed using *blue* bars, because blue is the default color for a *discrete* measures. For *continuous* measures, *green* is used instead, as exemplified by the *quantile chart* (*Cf.* Single Dependent Variable).



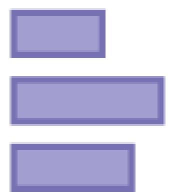
Blue is for discrete measures



Green is for continuous measures

Single Measure

If the single variable at hand is dependent and discrete, it could be drawn as a frequency chart rotated 90° clockwise so as to depict the variable on the vertical axis, and the count of its values on the horizontal axis. This chart could be called an **horizontal frequency chart**.

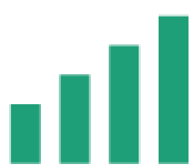


Horizontal Frequency Chart

Nevertheless, if the number of possible values for the variable is low, a **donut chart** would work.

Donut Chart

Finally, if the variable is dependent but continuous, its actual values can be depicted directly against the vertical axis through their quantiles by producing a **quantile chart**. For example, if real estate is limited, a quartile chart will show four vertical bars, with one bar for each quartile, and each bar showing the value of its respected quartile.



Quantile Chart

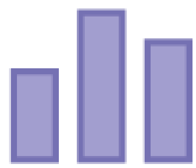
Summary of Primary Charts

In summary, the following primary charts can be produced for univariate datasets:

Dimension

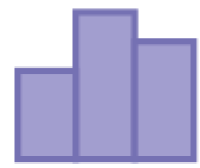
(Independent Variable)

Nominal



Frequency Chart

Incremental

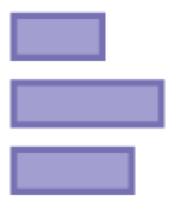


Histogram

Measure

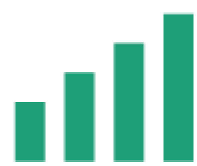
(Dependent Variable)

Discrete



Horizontal Frequency Chart

Continuous



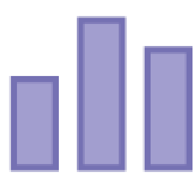
Quantile Chart

Special Primary Charts

Some typologies call for special primary charts. For example, a *denominational*, *identificational*, or *lexical* variable might have such a high cardinality (number of unique values) that a frequency chart might become unsuitable, even when restricted to the subset of values that have the highest cardinality. In such a case, further summarization of the data is necessary, and the omnipresent word cloud does not have the quantitative qualities that an experienced data analyst is entitled to expect — not to mention the fact that it is not a chart.

In this particular case, one might think of computing an additional pivot on the result of the pivot previously computed. The first pivot had the variable as sole dimension, and a *COUNT* as aggregation. The second pivot would have the first pivot's individual counts as sole dimension, and a *COUNT* as aggregation. In other words, it would be a count of counts, considering the first pivot's counts as values of an independent discrete variable (of *cardinal* typology).

This count of counts would indicate how many values of the original variable are found once, how many are found twice, how many are found three times, and so on and so forth. Interestingly, this count of counts is commonly used by natural language processing systems. Last but not least, the resulting **frequency of frequency chart** will conveniently depict proper unique keys (variables which values do not repeat themselves) as a single bar.



Frequency of Frequency Chart

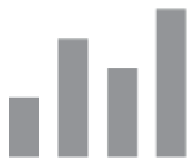
Secondary Charts

While *primary charts* apply to univariate datasets, **secondary charts** apply to bivariate datasets. For these three cases must be distinguished: one independent variables and one dependent variables, two independent variables, and two dependent variables. The first one is the most common, but the other two are of value as well. All three will be reviewed in details in order to introduce more and more advanced charts.

One Dimension and One Measure

Many sources credit William Playfair (circa 1780) with inventing the bar chart, however a Frenchman, Nicole Oresme used a bar chart in a 14th century publication, "The Latitude of Forms", to plot velocity of a constantly accelerating object against time. [Source: JPowered]

With such a backdrop and the consensus that line charts were introduced at least three centuries later (*Cf.* Line chart on Wikipedia), the **bar chart** could be considered as the very first chart to have been produced by proto statisticians.



Bar Chart

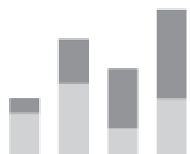
In its simplest form, a bar chart depicts a discrete independent variable on its horizontal axis, and a continuous dependent variable on its vertical axis. With that in mind, one might wonder whether bar charts are suitable for *any* kind of *discrete* independent variable, and *any* kind of continuous dependent variable.

For example, should one use different charts for *nominal* or *incremental* independent variables, or different charts for *intensive* or *extensive* dependent variables? All in all, should one consider four different charts for the four combinations of variables outlined above? And if so, is there a way to decide which charts to use for them?

Principia Pictura advocates for positive answers to all these questions, and provides a rational framework for deciding which type of chart should be used for each and every combination. Starting with the bar chart, one could wonder which combination of variables it is most suited to, then suggest incremental variations from it.

As mentioned earlier [*Cf.* Single Dimension], what distinguishes a *frequency chart* from a *histogram* is the horizontal padding between bars (or absence thereof), favoring the former for *nominal* variables and the latter for *incremental* ones. This suggest that a bar chart, which features an horizontal padding as well, is best suited to *nominal* independent variables.

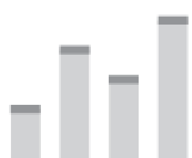
From there, one needs to decide whether a bar chart is more suited to *intensive* or *extensive* dependent variables. To answer this question, one could point to the **stacked bar chart**, which suggests that the dependent variable depicted against the vertical axis is summable and therefore *extensive*, since the bars depicting its values can be stacked on top of each other.



Stacked Bar Chart

In conclusion, a bar chart is most suited to a *nominal* independent variable combined with an *extensive* dependent variable. From there, one might wonder which chart would be most suited to an *intensive* dependent variable instead.

For this scenario, the ideal chart should convey the notion that values of the dependent variable cannot be summed, therefore should not make use of stackable bars. Nevertheless, this chart should also be visually similar to the bar chart, especially with respect to their common *nominal* independent variable. This suggests the use of a **level chart**.



Level Chart

While a *level chart* looks similar to a *bar chart*, it is made of *levels*, not *bars*. These levels aim to show the local value of an *intensive* variable, instead of the global extent of an *extensive* variable. Underneath a level, the bars should be rendered with transparency, a lighter shading, or a lesser color saturation, so as to give the impression of a trajectory for the level. As a result, a level chart should give a clear indication that unlike bars, levels cannot be stacked, therefore conveying the notion that the depicted dependent variable is not summable, hence *intensive*.

Looking at the independent variable now, one might wonder which chart should replace the level chart when going from a *nominal* variable to an *incremental* one. Because an *incremental* independent variable calls for some concept of continuity on the horizontal axis, and because the combination of variables being considered includes an intensive therefore non-summable one to be depicted on the vertical axis, an obvious candidate is the conventional **line chart**.

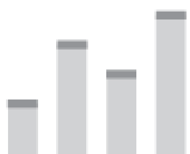
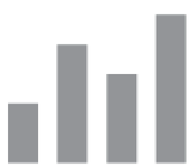
Line Chart

A line chart is particularly well suited to this combination of variables because a line drawn between two points conveys the notion of linear interpolation that could be performed should one decide to subdivide the increments of the *incremental* independent variable depicted against the horizontal axis, and to interpolate corresponding values of the *intensive* therefore *continuous* dependent variable depicted against the vertical axis. Finally, a line chart clearly conveys the notion of non-summability of the *intensive* dependent variable since lines cannot be stacked.

The last combination that needs to be considered is when the *intensive* dependent variable visualized by a line chart is replaced by an *extensive* one. In that case, the line chart can be replaced by an **area chart**, since areas can be stacked, therefore properly convey the notion of summability of the *extensive* dependent variable depicted against the vertical axis.

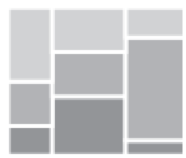
Area Chart

In summary, all four combinations of variables can be visualized with conventional charts, according to simple conventions governing the suggestion of continuity or discontinuity alongside the horizontal axis, and the use of stackable or non-stackable marks to depict summable or non-summable dependent variables against the vertical axis. The following table outlines these conventions, with the term **dimension** used as alias for *independent* variable, and **measure** as alias for *dependent* variable.

	Nominal Dimension	**Incremental Dimension**
Intensive Measure	 Level Chart	 Line Chart
Extensive Measure	 Bar Chart	 Area Chart

Two Dimensions

A single independent variable is depicted with a frequency chart or a histogram produced from the result of a pivot defined with the variable as sole dimension and *COUNT* as aggregation. Similarly, two independent variables require some kind of frequency chart that could depict the result of a pivot defined with the two variables as dimensions and *COUNT* as aggregation. Fortunately, such a chart exists: it is called a **mosaic plot** and was introduced in 1981 by Hartigan and Kleiner, then expanded on by Friendly in 1994. [Source: Wikipedia]



Mosaic Plot

When produced for two independent variables, every rectangle of a mosaic plot depicts a pair of discrete values for the two variables, and its relative area is proportional to the pivot count of these two values. In order to visualize the relative contributions the two variables, the values of one of them is depicted through different colors, while the values of the other is depicted through different columns of rectangles. As a result, this visualization technique is capable of conveying a lot of information through a very small amount of real estate.

Interestingly, mosaic plots can be produced to visualize a virtually unlimited number of independent variables, even though they become much less legible beyond two variables. Traditionally, mosaic plots visualizing just two independent variables have been called **Marimekko diagrams** or **Mekko charts**, due to their resemblance to a Marimekko print.

It should also be noted that a mosaic plot is produced by subdividing a rectangular area across two or more dimensions. While this subdivision technique can be used to produce a standalone chart, it can also be used to subdivide the *rectangle* marks of other charts like bar charts in order to visualize additional independent variables (*Cf.* Mark Subdivision).

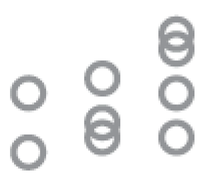
Two Measures

While the visualization of a single *dependent* variable relied on the computation of a pivot (*COUNT* for a discrete variable, *QUANTILE* for a continuous one), the visualization of two *dependent* variables can be performed on the raw data quite effectively. To do so, three cases need to be distinguished: two *continuous* variables, one *discrete* and one *continuous*, and two *discrete* variables. The most common of the three is the first one.

The visualization of two *continuous* dependent variables is best achieved with a **scatter plot**. This chart is quite effective because it facilitates the visual detection of clusters, or can be used to visualize clusters instead of raw data when the number of individual data points is too large. It is also perfectly suited to the overlay on top of a *geographic map* when the two *dependent* variables represent a pair of latitude and longitude.

Scatter Plot

When one of the two *dependent* variables becomes *discrete*, the *scatter plot* can be replaced by a **circle plot**, which uses *circle* marks instead of *dots* and organizes them in discrete columns alongside the horizontal axis. The use of *circles* marks instead of *dots* is motivated by the fact that marks on a *circle plot* are more likely to overlap with each other, hence improves legibility.



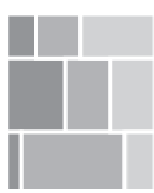
Circle Plot

Finally, when the two *dependent* variables are *discrete*, a pivot with *COUNT* aggregations across the two *dependent* variables used as dimensions becomes not only necessary but fundamentally equivalent to the outline of the raw data, since no information is lost in the process. And the most suitable primary chart to visualize the results of this pivot is a **horizontal mosaic chart**, which displays rows of *rectangles* instead of columns. This is consistent with the use of a *frequency chart* for a single nominal independent variable and a *horizontal frequency chart* for a single *discrete* dependent variable.

If the cardinalities of the two *dependent* variables are too large, an additional pivot is required, thereby computing a count of counts similar to the one displayed by a *frequency of frequency chart*. In this case, a *horizontal mosaic* chart would still be used, but every *rectangle* would display a count of counts instead of a simple count. In order to distinguish counts from counts of counts, the former could use different colors for distinguishing values of the second *dependent* variable, while the latter could use different saturations of the same color (green by default).

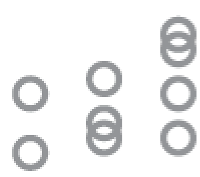
In summary, two *dependent* variables can be visualized in the following fashion:

2 discrete variables



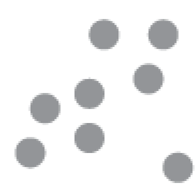
Horizontal Mosaic Chart

1 discrete, 1 Continuous



Circle Plot

2 Continuous Variables



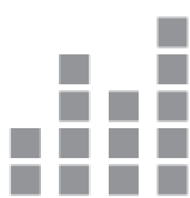
Scatter Plot

Tabular Secondary Charts

The basic secondary charts recommended for one dimension (independent variable) and one measure (dependent variable) all focused on *continuous* measures, which are the most common. But what about secondary charts for *discrete* measures? These can be produced as well, using a family of charts called **tabular**, in contrast to the family of charts called **cartesian** used for continuous measures (the *cartesian* and *tabular* terms were introduced by *Principia Pictura*).

Because *discrete* measures are much less common than *continuous* ones, *tabular* charts are much less common than their *cartesian* counterparts as well. In fact, it is quite possible that some tabular charts used by *Principia Pictura* are original creations or accidental discoveries, which is why some of them have rather unusual names. It is also quite possible that such charts have been produced in the past under different names. If that is the case, the reader is invited to contact the author[3] so that due credit could be given back.

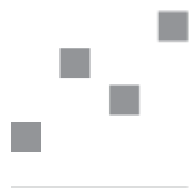
The tabular counterpart of the bar chart is called a **dot chart**. This chart simply replaces solid bars by stacks of discrete dots. In theory, dots used in dot charts could have any shape, but square dots should be preferred to any alternative, for they provide the highest level of legibility, without introducing any comparative bias. Solid circles could be used as well, but reduce legibility, especially on smaller sizes, and icons or symbols should be avoided at all costs. Lastly, dot charts should not be confused with dot plots, which have a very specific statistical meaning.



Dot Chart

[3] Questions and comments should be sent to ismael@stoic.com.

The tabular counterpart of the level chart is called a **bit chart**. This chart simply replaces a continuous level by a discrete bit visualized with a *square* mark. For such a chart, it is worth noting that the order of the rows on which these bits are displayed depends on the discrete typology of the dependent variable depicted against the vertical axis. If this variable supports sorting (lexical, sequential, ordinal, incremental, cardinal), the order is meaningful. Otherwise, it is not, and the order within which values of the dependent variable are depicted against the vertical axis could be randomly defined in order to visually convey this fact.

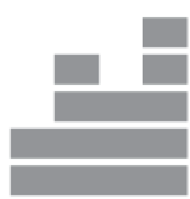


Bit Chart

The tabular counterpart of the line chart is called a **band chart**. This chart somehow replaces straight lines by horizontal bands visualized with *rectangle* marks connected together through thin vertical lines. The quantum leaps depicted by the connection lines are used to visually convey the continuity of the horizontal axis (since the visualized dimension is continuous), and the discontinuity of the vertical axis (since the visualized measure is discrete).

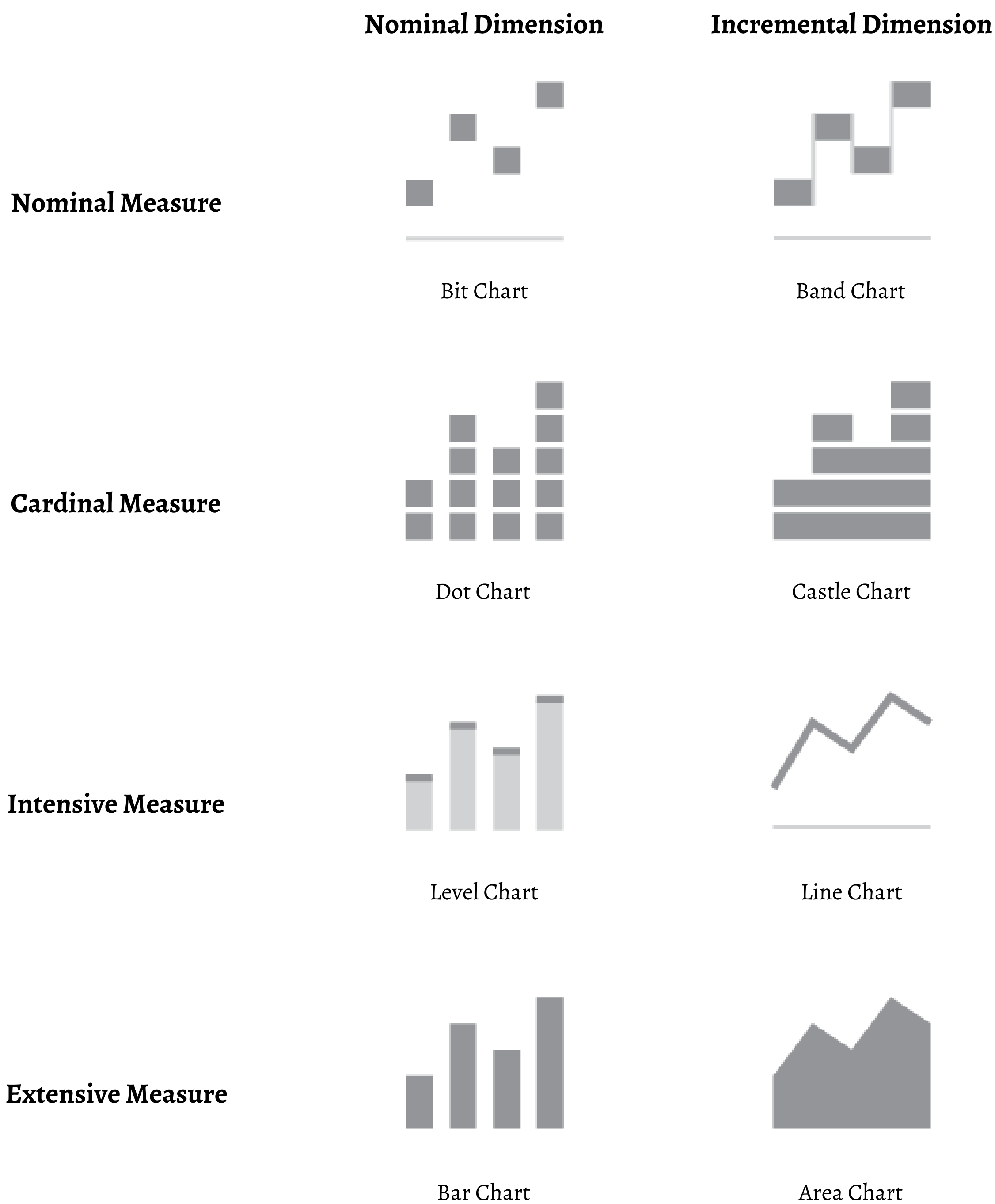
Band Chart

Finally, the tabular counterpart of the area chart is called a **castle chart**.



Castle Chart

In describing the tabular counterparts to basic cartesian charts, the *intensive vs. extensive* dichotomy used to classify cartesian charts was voluntarily omitted. Nevertheless, this dichotomy is essential, and so is its tabular counterpart, which is *nominal vs. cardinal*.

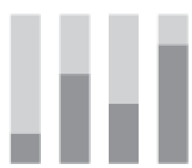


Absolute Secondary Charts

According to *Principia Data*, the values of *numerical* variables can be bounded with lower and upper limits. The typologies for such variables are called **absolute**, and absolute counterparts can be defined for every numerical typology, as summarized on the following table:

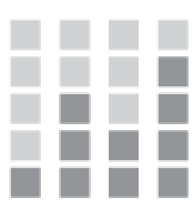
	Non-Summable	**Summable**
Discrete	Incremental → *Quantile*	Cardinal → *Fractional*
Continuous	Intensive → *Rational*	Extensive → *Percent*

Consequently, *absolute* counterparts must be defined for every basic secondary chart. To do so, the full range of possible values is outlined with transparency, a lighter shading, or a lesser color saturation than the shading or saturation used for depicting actual values. For example, the absolute counterpart of the bar chart would be an **absolute bar chart**.



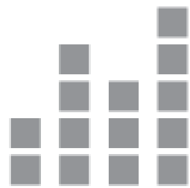
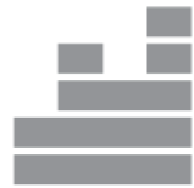
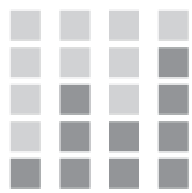
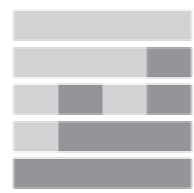
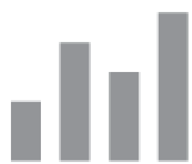
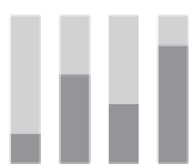
Absolute Bar Chart

This chart should not be confused with the **absolutized stacked bar chart**, which is used to display multiple series of the same absolute dependent variable, even though they look similar, and their visual proximity is somehow justified. The **absolute dot chart** follows a similar logic, and so do the absolute counterparts to the *line chart* and *area chart*.



Absolute Dot Chart

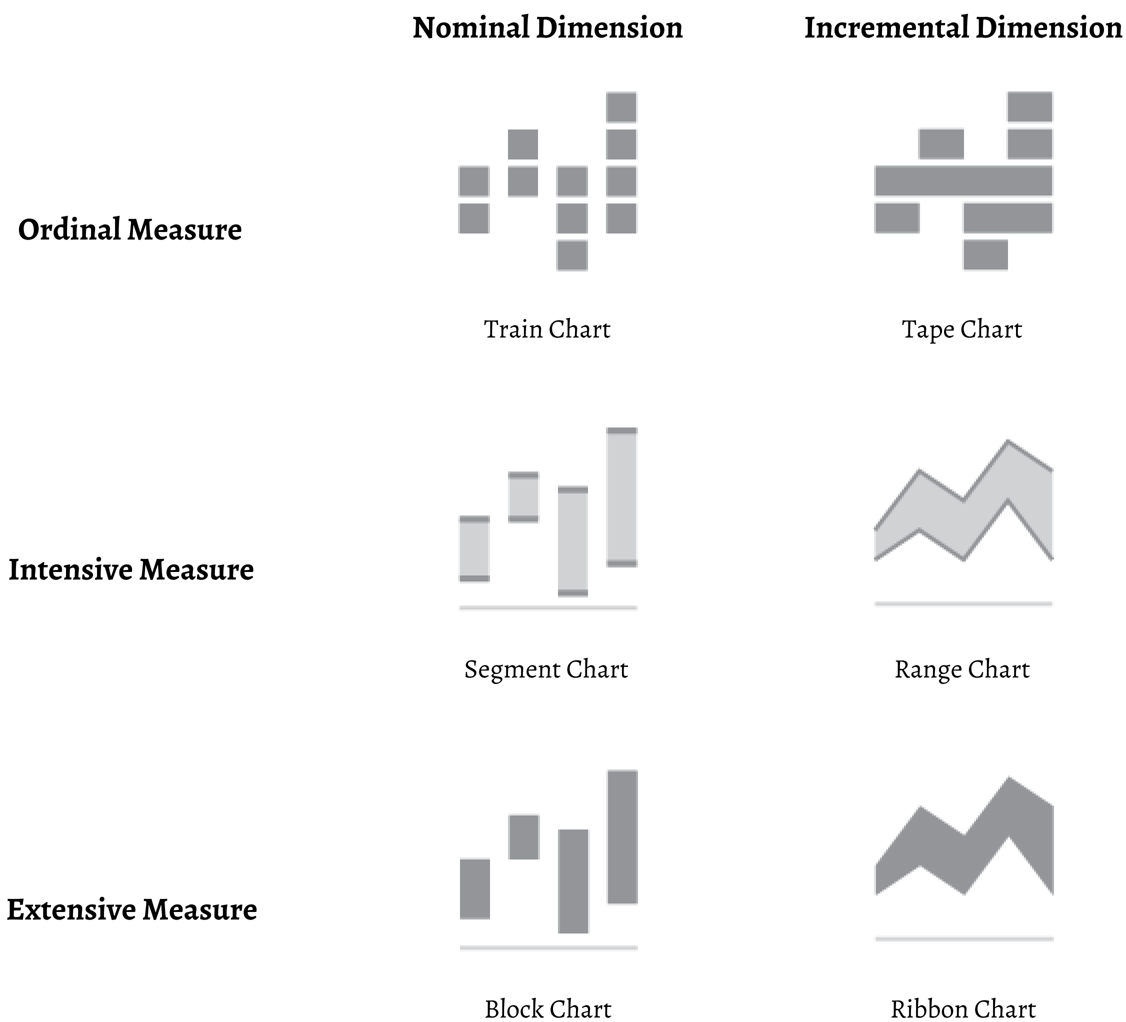
The set of absolute counterparts for secondary charts is outlined on the following table, using the *fractional* typology for absolute discrete measures, and the *percent* typology for absolute continuous measures. Equivalent charts for the *quantile* and *rational* have yet to be defined[4].

	Nominal Dimension	**Incremental Dimension**
Cardinal Measure	 Dot Chart	 Castle Chart
Absolute Measure (Fractional Typology)	 Absolute Dot Chart	 Absolute Castle Chart
Extensive Measure	 Bar Chart	 Area Chart
Absolute Measure (Percent Typology)	 Absolute Bar Chart	 Absolute Area Chart

[4] This definition could be the subject of a future revision to *Principia Pictura*.

Differential Secondary Charts

So far, secondary charts have been used to visualize an independent variable in relation to a single dependent variable, but there are some cases where an independent variable must be visualized in relation to a pair of interrelated dependent variables. This happens when one needs to visualize the differences between two variables of the same datatype, or the differences between two values of the same variable. The charts used to visualize these datasets are called **differential charts**, and they are defined for the *ordinal*, *intensive*, and *extensive* typologies.



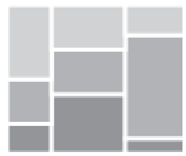
Tertiary Charts

Next, *Principia Pictura* offers a set of **tertiary charts** for datasets made of three variables:

- Three or More Dimensions
- Two Dimensions and one Measure
- Two Measures and one Dimension
- Three or More Measures

Three or More Dimensions

As mentioned earlier (*Cf.* Two Dimensions), the most suitable chart for the visualization of two or more dimensions (and therefore three or more dimensions) is the **mosaic chart**, even though its legibility for more than two variables is questionable.



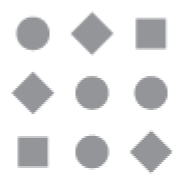
Mosaic Plot

Two Dimensions and One Measure

Different tertiary charts can be used to visualize a dataset made of two dimensions and one measure, depending on the typology of the measure. For these charts, the two dimensions are assumed to be *nominal*, *lexical*, *sequential*, *ordinal*, or *incremental*.

Nominal Measure

If the measure is *nominal*, the deliciously-named **chocolate chart** can be used. This chart makes use of *symbol* marks for depicting the discrete measure, which limits its applicability to datasets which discrete measure has a low cardinality (10 or less typically).

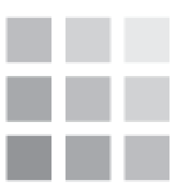


Chocolate Chart

Ordinal Measure

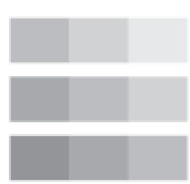
If the measure is *ordinal*, three tertiary charts are available, depending on the typologies of the two dimensions. If both dimensions are *nominal*, a **heatmap** can be used. If one of the two dimensions is *incremental*, a **gradient chart** will be perfectly suited. And if both dimensions are *incremental*, a **contour plot** will be prefered.

2 Nominal Dimensions



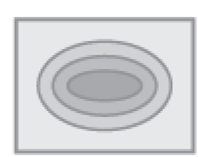
Heatmap

1 Incremental Dimensions



Gradient Chart

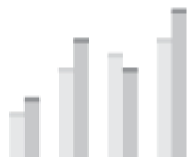
2 Incremental Dimensions



Contour Plot

Intensive Measure

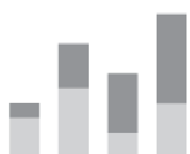
If the measure is *intensive* and the cardinalities of the dimensions are low, a *level chart* with a *cohort* (*Cf.* Cohorts) for the dimension of lowest cardinality will be prefered to a *heatmap*, for it will make it easier to compare values of the intensive measure across the two dimensions.



Level Chart with Cohort

Extensive Measure

If the measure is *extensive* and the cardinalities of the dimensions are low, a *bar chart* with a cohort for the dimension of lowest cardinality could be used as well, but a **stacked bar chart** will be usually prefered, for it is better at conveying the summability of the *extensive* measure, and it provides an accurate depiction of aggregations of the measure summed by the first dimension. From a technical standpoint, a *stacked bar chart* is actually a *bar chart* which *rectangle* marks have been subdivided across one dimension (*Cf.* Details).



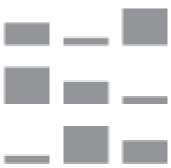
Stacked Bar Chart

This example illustrates the complexity of defining heuristics aimed at recommending the most suitable chart for a given datasets (*Cf.* Chart Recommendations). In this particular instance, both options are equally valid, and the decision to use one versus the other is entirely driven by the balance of two conflicting requirements: comparing values of the measure, or summing them across the first dimension. In most cases, only a human could make this decision.

Absolute Measure

If the measure is *absolute*, three tertiary charts are available, depending on the typologies of the two dimensions. If both dimensions are *nominal*, a **scale chart** can be used. If one of the two dimensions is *incremental*, a **stair chart** will be perfectly suited. And if both dimensions are *incremental*, a **contour plot** will be prefered, as is the case with an ordinal measure.

2 Nominal Dimensions

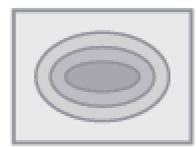


Scale Chart

1 Incremental Dimensions

Stair Chart

2 Incremental Dimensions



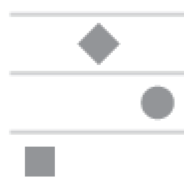
Contour Plot

One Dimension and Two Measures

Different tertiary charts can be used to visualize a dataset made of one dimension and two measures, depending on the typology of the measure. For these charts, the two dimensions are assumed to be *nominal*, *lexical*, *sequential*, *ordinal* only (the *incremental* case is ignored for now[5]).

Two Discrete Measures

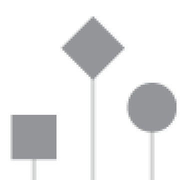
If the two measures are *discrete*, a **spot chart** is recommended. The *spot chart* is very similar to the *bit chart*, but uses variable *symbol* marks instead of constant *square* marks. The use of symbol marks explains why an *incremental* counterpart of this chart is not obvious, unlike the *band chart*, which is an *incremental* counterpart of the *bit chart* (*Cf.* Tabular Secondary Charts).



Spot Chart

One Discrete and One Continuous

If one measure is *discrete* and the other is *continuous*, the other deliciously-named **lollipop chart** is recommended. This chart is derived from the *spot chart* and replaces the discrete vertical axis with a continuous one by using a vertical *line* mark as a way to show the vertical trajectory of *symbol* marks, much like a *level chart* (*Cf.* One Dimensions and One Measure).

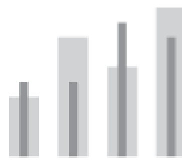


Lollipop Chart

[5] This case could be the subject of a future revision to *Principia Pictura*.

Two Intensive Measures

If the two measures are *intensive*, a **bullet chart** is recommended. This chart is actually the *superimposition* of two level charts (*Cf.* Chart Superimposition), one for each measure. As such, it requires a decision to be made with respect to which variable is displayed underneath (*target*), and which is displayed on top (*actual*).



Bullet Chart

One Intensive and One Extensive

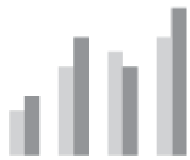
If one measure is *intensive* and the other is *extensive*, a **bar and line chart** is recommended, as meteorologists plotting temperatures (*intensive*) and precipitations (*extensive*) have known for a long time (*Cf.* Climograph on Wikipedia). This chart is actually the *superimposition* of a *line chart* and a *bar chart* (*Cf.* Chart Superimposition), with the former always displayed on top.

Bar and Line Chart

This chart is often called a *combo chart*, but the *bar and line chart* denomination should be prefered, since the *combo* (short for *combination*) term might be confused with the *composition* term (*Cf.* Chart Composition), and composited charts are not limited to the superimposition of a *line chart* on top of a *bar chart*. It should also be noted that unlike the *bullet chart*, this chart requires two separate vertical axes defined against two incompatible typologies. As such, the comparison of relative variations across the two measures makes sense, but the comparison of absolute values does not. Therefore, this often abused chart should be used with caution.

Two Extensive Measures

Finally, if the two measures are *extensive*, a *bar chart* with a *cohort* (*Cf.* Cohorts) is recommended. Nevertheless, this visualization is possible only if the two measures are *congruent*, meaning that they should be able to share a common vertical axis. In other words, they must have the same unit of measurement. Interestingly, the same is true for the *bullet chart* used to visualize one dimension and two *intensive* measures (*Cf.* Two Intensive Measures).



Bar Chart with Cohort

Three or More Measures

If three or more measures must be visualized without any dimension, the **parallel coordinates** are recommended, even though their legibility on limited real estate is questionable. Each and every measure is visualized through a vertical bar. If a measure is discrete, its vertical bar can be segmented and labelled or colored in order to properly distinguish and identify the measure's individual values, even though such a refinement will make the chart even less legible.

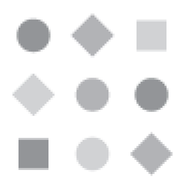
Parallel Coordinates

Quaternary Charts

Finally, *Principia Pictura* offers a set of **quaternary charts** for datasets made of four variables. Beyond that, the combinatorics of typologies become too large for the framework to remain effective, and predefined recommendation must be replaced by recommendation algorithms. For these charts, the two dimensions are assumed to be *nominal*, *lexical*, *sequential*, *ordinal* only (the *incremental* case is ignored for now[6]).

Two Discrete Measures

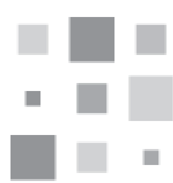
If the two measures are *discrete*, yet another deliciously-named **bonbon chart** is recommended. This chart is derived from the *chocolate chart* (*Cf.* Nominal Chart), and adds a second discrete measure to it by coloring the *symbol* marks.



Bonbon Chart

One Discrete and One Continuous

If one measure is *discrete* and the other is *continuous*, the **tile chart** is recommended. This chart is derived from the *bonbon chart*, and replaces symbols by sizes and uses a *square* mark.

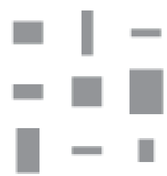


Tile Chart

[6] This case could be the subject of a future revision to *Principia Pictura*.

Two Continuous Measures

Finally, if the two measures are *continuous*, a **screen chart** is recommended. This chart is derived from the tile chart, and replaces colors and sizes by heights and widths, while using a *rectangle* mark instead of a *square* mark in order to control heights and widths independently.



Screen Chart

Chart Axes

The **axes** of a chart define the number and typologies of the variables that can be visualized through it. As such, they should not be confused with its **marks**, which define how these variables are visualized through the chart. For example, a chart's mark is often defined by multiple axes such as *width*, *height*, or *color*.

Similarly, the **axes** of a chart should not be confused with its **scales**. The scales of a chart define how the chart is rendered on the two-dimensional canvas of a screen or paper, with optional support for depth rendering. The basic scales are *horizontal*, *vertical*, and *depth*. In addition, scales for *color*, *symbol*, or various *sizes* can be defined. A scale is always associated to one and only one axis. Therefore, when referring to the *horizontal axis* or to the *vertical axis*, this paper actually refers to the axes that are associated to the *horizontal* and *vertical* scales respectively.

Axes are usually defined with the following properties:

- **Name** and **Alias** (name of the axis in a polar system — *Cf.* Chart Polarization)
- **Relation** (*independent* or *dependent*)
- **Typology** (minimum typology of a variable that can be bound to the axis)
- **Signable** (whether a signed variable can be bound to the axis)
- **Required** (whether the axis is required)
- **Multiple** (whether multiple variables can be bound to the same axis)
- **Cohorts** (whether the axis supports cohorts)
- **Details** (whether the axis supports details)
- **Columnar** and **Tabular** (whether the axis is columnar or tabular)
- **Latitudinal** and **Longitudinal** (whether the axis is latitudinal or longitudinal)
- **Altitudinal** (whether the axis is altitudinal)
- **Differential** (whether the axis is differential)

Axis Relation

An axis is also defined with a **relation**, which is either *independent* or *dependent*. An *independent* variable can only be bound to an *independent* axis, and a *dependent* variable can only be bound to a *dependent* axis. Most charts are defined with a single *independent* axis, while some have none or two, and *Principia Pictura* does not define any standard chart with more than two. Nevertheless, the following mechanisms can be used to bind more *independent* variables to chart axes:

- **Axis Multiplicity** (*Cf.* Axis Multiplicity)
- **Cohorts** (*Cf.* Cohorts)
- **Details** (*Cf.* Details)

Furthermore, additional mechanisms can be used to visualize additional *independent* variables:

- **Chart Tabulation** (*Cf.* Chart Tabulation)
- **Chart Juxtaposition** (*Cf.* Chart Juxtaposition)
- **Chart Superimposition** (*Cf.* Chart Superimposition)
- **Mark Subdivision** (*Cf.* Mark Subdivision)

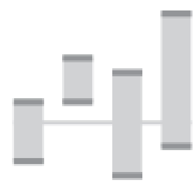
Axis Typology

An axis is defined with the minimum **typology** of a variable that can be bound to the axis, either as an actual typology (*e.g. intensive*) or as a group of typologies (*e.g. discrete, summable, etc.*). In this context, *minimum* means that any typology having all the mathematical operators granted to the *minimum typology* is applicable as well. *Principia Pictura* defines the following typological groups:

- **Absolutizable** — *Incremental* (or better) OR *Intensive* (or better)
- **Absolutization** — *Incremental* OR *Intensive*
- **Any** — Any typology
- **Continuous** — Any *continuous* typology
- **Discrete** — Any *discrete* typology
- **Discretizable** — Any typology
- **Historical** — *Chronological* OR *Temporal*
- **Quantizable** — *Sequential* (or better) OR *Continuous*
- **Sectorizable** — *Sectorial* (or better) OR *Directional* (or better)
- **Summable** — *Cardinal* (or better) OR *Extensive* (or better)
- **Summation** — *Cardinal* OR *Extensive*

Signable Axis

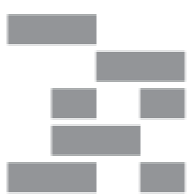
Some axes are **signable**, meaning that *signed* variables can be bound onto them. For example, the *horizontal* and *vertical* axes of the *bar chart* are *signable*, but its *width* axis is not. Similarly, the *horizontal*, *baseline*, *endline*, and *gain* axes of the segment chart are all *signable*.



Signed Segment Chart

It should be noted that all *color* axes are *signable*, thanks to the support of *diverging color palettes*. (*Cf.* Diverging Color Palette). In a similar fashion, *symbol* axes could be made *signable* by associating a signed sentiment index to each and every symbol. For example, an arrow pointing up would have a positive index, while an arrow pointing down would have a negative one.

In contrast, length-related axes such as *length*, *width*, and *size* axes are never *signable*. Similarly, axes bound to *relational* variables like the *parent* and *predecessor* axes of the *gantt chart* or the *source* and *target* axes for the *arc chart* are never *signable*.



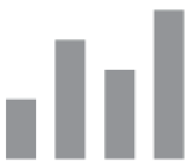
Gantt Chart

Arc Chart

The *signable* property of a chart axis is important because it restricts the set of axes onto which *signed* variable can be bound. It also instructs the *chart rendering engine* that a **zero reference line** or a signed legend (*e.g. diverging color palette*, *diverging symbol set*) should be depicted in order to properly depict these variables and their signed nature.

Required Axis

Some chart axes are optional, while others are **required**. For example, the *color* axis of a *bar chart* is optional, but its *horizontal* axis is required. Sometimes, only one of multiple optional axes is required. For example, with a *differential chart* like *block*, *segment*, or *range*, the *baseline* is required, but only one of the *endline* or *gain* is required. This means that a *differential chart* must be defined either with a *baseline* and *endline*, or with a *baseline* and *gain*.



Bar Chart

Block Chart

Most charts are defined with one or two required axes. Following a principle of parsimony, *Principia Pictura* tries to keep the number of standard charts as small as possible and adds a potentially large number of optional axes to its standard charts. For example, the *scatter plot* is defined with a total of eight axes (*horizontal*, *vertical*, *angle*, *length*, *size*, *color*, *symbol*, *value*).

Scatter Plot

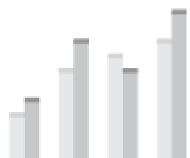
As a result, it might not be immediately obvious to the casual reader how some conventional charts can be produced by *Principia Pictura*. Nevertheless, the author of this paper has tried to support as many charts and statistical plots (*Cf.* Conventional Statistical Plots) with the current framework. Should any conventional chart or plot appear to fall out of the framework's envelope, the reader is invited to contact the author[7].

[7] Questions and comments should be sent to ismael@stoic.com.

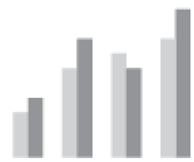
Axis Multiplicity

Some *dependent* chart axes can be repeated multiple times. For example, the *vertical* axes of the *line, level*, and *bar* charts can be duplicated on the left and on the right, allowing two separate measures to be displayed simultaneously against the *vertical* axis.

Multiple Line Chart



Multiple Level Chart



Multiple Bar Chart

The multiplicity of a *dependent* axis should not be confused with the repetition of a mark associated to an *independent* axis with a *cohort* (*Cf.* Cohorts). While the two might lead to visually identical charts, the former allows multiple *dependent* variables to be bound to the same axis, while the latter allows an *independent* variable to be visualized through the replication of a mark. These are two fundamentally different concepts that produce similar visualizations for two fundamentally different datasets.

While most chart axes that support **axis multiplicity** can only be repeated twice, some axes can be repeated an unlimited number of times, thereby supporting the binding of an unlimited number of *dependent* variables. Such is the case for the *coordinates* axis of *parallel coordinates*.

Parallel Coordinates

Finally, *independent* axes do *not* support *axis multiplicity* (they support *cohorts* instead).

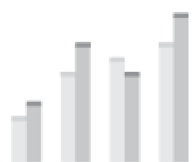
Cohorts

cohort

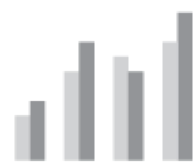
: an ancient Roman military unit, comprising six centuries, equal to one tenth of a legion.

For some charts, one *independent* axis can support the replication of the mark it is associated to, in relation to an additional *independent* variable called a **cohort**. For example, the *lines* on a *line chart*, the *levels* on a *level chart* or the *bars* on a *bar chart*.

Line Chart with Cohort



Level Chart with Cohort



Bar Chart with Cohort

The repetition of a mark associated to an *independent* axis should not be confused with the multiplicity of a *dependent* axis (*Cf.* Axis Multiplicity). While the two might lead to visually identical charts, the former allows an *independent* variable to be visualized through the replication of a mark, while the latter allows multiple *dependent* variables to be bound to the same axis. These are two fundamentally different concepts that produce similar visualizations for two fundamentally different datasets.

Much like a Roman cohort was comprised of six centuries, a *cohort* should be limited to *independent* variables that have a low cardinality. Since the members of a *cohort* are usually distinguished through different colors, the introduction of a cohort requires the use of a color palette, and *nominal* color palettes cannot have more than 20 colors (*Cf.* Nominal Color Palette).

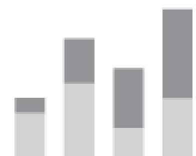
Finally, *dependent* axes do *not* support *cohorts* (they support *axis multiplicity* and *details* instead).

Details

For some charts, one *dependent* axis can support the detailing of the mark it is associated to, in relation to an additional *independent* variable called a **detail**. For example, the *line* on a *line chart*, the *areas* on an *area chart*, or the *bars* on a *bar chart*, or.

Line Chart with Detail

Area Chart with Detail



Bar Chart with Detail

For some marks like *line*, detailing consists in replicating the mark as many times as there are values for the *detail* variable. For other marks like *area*, detailing consists in splitting the area into as many splits as the *detail* variable has values — in fact, one could consider the replication of a *line* mark as a special case of *splitting*, by considering the *line* as an area which width is *null*. And for marks like *rectangle*, detailing consists in subdividing the mark into as many subdivisions as the *detail* variable has values (*Cf.* Mark Subdivision).

Detailing by *replication* or *splitting* can be done with only one *detail* variable, while detailing by *subdivision* can be done with as many *detail* variables as supported by the subdivision technique: the *treemap* and *partition* subdivisions support a single dimension, but the *mosaic* subdivision supports an unlimited number of them (2 being the practical maximum though).

Finally, *independent* axes do *not* support *details* (they support *cohorts* instead).

Columnar and Tabular Axes

Most *cartesian charts* (*Cf.* Cartesian Charts) are defined with *horizontal* and *vertical* axes, and most *tabular charts* (*Cf.* Tabular Chart) are defined with *column* and *row* axes (or equivalent axes). *Principia Pictura* attaches many specific behaviors to these axes. For example, the ability to swap them in order to control a chart's orientation (*horizontal* or *vertical* — *Cf.* Chart Orientation).

Some of these behaviors are dependent upon the *discrete* nature of these axes, as defined by their ability to be bound to *discrete* variables. For these reasons, some charts are defined with a *columnar* axis (but never more than one) and/or a *tabular* axis (but never more than one).

A *columnar* axis is a *discrete* counterpart to a *longitudinal* axis.

A *tabular*[8] axis is a *discrete* counterpart to a *latitudinal* axis.

Note to the reader: these advanced concepts are certainly quite technical and only matter to the implementer of some *chart rendering engine*. They have limited relevance to the end-user of such a tool, and can be ignored in a first reading of this paper (or forever, really).

[8] *Tabular* is the adjective for *row*, much like *columnar* is the adjective for *column*.

Latitudinal, Longitudinal, and Altitudinal Axes

Most *cartesian charts* (*Cf.* Cartesian Charts) are defined with *horizontal* and *vertical* axes, and most *tabular charts* (*Cf.* Tabular Chart) are defined with *column* and *row* axes (or equivalent axes). *Principia Pictura* attaches many specific behaviors to these axes. For example, the ability to swap them in order to control a chart's orientation (*horizontal* or *vertical* — *Cf.* Chart Orientation).

Some of these behaviors are dependent upon the *continuous* nature of these axes, as defined by their ability to be bound to *continuous* variables. For these reasons, some charts are defined with a *longitudinal* axis (but never more than one) and/or a *latitudinal* axis (but never more than one). Some charts are also defined with an *altitudinal* axis, which is used to suggest a third dimension on a two-dimensional canvas. An iconic example is the *depth* axis of a *line chart*, which is used to produce a joy division chart to great effect.

A *longitudinal* axis is a *continuous* counterpart to a *columnar* axis.

A *latitudinal* axis is a *continuous* counterpart to a *tabular*[9] axis.

An *altitudinal* axis has no known *discrete* counterpart.

Note to the reader: these advanced concepts are certainly quite technical and only matter to the implementer of some *chart rendering engine*. They have limited relevance to the end-user of such a tool, and can be ignored in a first reading of this paper (or forever, really).

[9] *Tabular* is the adjective for *row*, much like *columnar* is the adjective for *column*.

Differential Axes

Differential charts visualize the differences between two measures of the same datatype or two values of the same measure (*Cf.* Differential Secondary Charts). For example, a *block chart* can be used to visualize the differences between two *extensive* measures across a *nominal* dimension, while a *ribbon chart* does the same across an *incremental* dimension.

Block Chart

Ribbon Chart

Differential charts are always defined with three axes: *baseline*, *endline*, and *gain*. The *baseline* is always required, alongside either *endline* or *gain* (one and only one required). Values of these three axes respect the following equation:

$$baseline + gain = endline$$

While such charts could have been defined with only a *baseline* and *endline* or a *baseline* and *gain*, offering both options makes it easier to bind the measures of existing datasets onto their axes without having to compute *ad hoc* measures just for the purpose of visualization.

Axis Scale

A chart's axis is an abstract concept that is not visualized directly on the chart. Instead, it is visualized through its **scale**. For example, the horizontal **rule** usually displayed at the bottom of a *line chart* is the *horizontal* scale corresponding to the *line chart*'s *horizontal* axis.

Line Chart with Scale

Because there exists a one-to-one relationship between an *axis* and its *scale*, the two are often confused with each other, but the reader is invited to make a serious attempt at considering them independently from each other, for it will make it easier to understand their respective properties, especially when using advanced visual enhancements (*Cf.* Visual Enhancements).

It is also worth noting that not all scales look like rules. For example, the scales for *color* or *symbol* axes are usually depicted through **legends**. In other words, a *scale* is an abstract concept as well, which is probably best understood through the following hierarchy:

Variable (*dimension* or *measure*)

bound to...

Axis (*independent* or *dependent*)

scaled with...

Scale

visualized through...

Rule or **Legend**

Axis Drill-Down

The process of selecting a value on the scale of a chart axis and producing another chart as a result is called **axis drill-down**. For a single value to be selected, the axis must be *discrete*. If a range of values needs to be selected, **axis brushing** is used instead (*Cf.* Axis Brushing). A similar process exists for marks (*Cf.* Mark Drill-Down).

Axis and mark drill-down and brushing are important processes, not just for the level of interactivity they confer to otherwise static visualizations: they also help the data analyst develop a deeper understanding of charts, axes, dimensions, and measures. And they can be used as teaching exercises, asking students which chart would be produced when selecting a value on a scale or a mark on the canvas.

Selecting a value from the scale of an axis is equivalent to filtering out all other possible values for the variable bound to the axis. Doing so essentially removes the variable from the dataset being visualized and makes the axis used to visualize it irrelevant. As a result, another chart must be produced, from the set of variables used to produce the original chart, minus the variable that was removed, and this chart will require one less axis than the original chart.

The removal of a variable from the dataset has different consequences depending on whether the variable is *independent* (*dimension*) or *dependent* (*measure*). A solid understanding of this dichotomy is essential to the proper use of drill-down techniques.

If the removed variable is *independent*, it means that the dataset visualized by the original chart had at least one *independent* variable, therefore was the result of a group-by aggregation (*pivot*).

If two or more *independent* variables were visualized by the original chart, removing one leads to a focus on the remaining ones, and the new chart will visualize the same *measures* as the original one, but one less *dimension*, hence will require one less *independent* axis.

But if a single *independent* variable was visualized by the original chart, the removal of this last dimension leads to a new dataset only made of measures, and the visual comparison of multiple measures independently of any dimension is of limited value. In such a context, the removal of the last *independent* variable is usually followed by an outlining of the source data that was aggregated by the pivot in the first place. This is where the *drill-down* term comes from, and the source data itself can be visualized through a chart, especially if its number of measures is small (*Cf.* Single Measure, Two Measures, Three or More Measures).

If the removed variable is *dependent*, two cases must be distinguished, depending on whether the dataset visualized by the original chart is raw data or aggregated data. In the case of raw data, the new chart will simply visualize one less *dependent* variable than the original one. In the case of aggregated data, if two or more *dependent* variables were visualized by the original chart, one less *dependent* variable will be visualized by the new one. But if a single *dependent* variable was visualized by the original chart, the removal of this last measure leads to a new dataset only made of dimensions. In such a context, the new chart will display *COUNT* aggregations, instead of the specific aggregations that could have been defined for the removed measure.

The axis drill-down behaviors outlined above can be summarized as follows:

Dimension		**Measure**		
Aggregated Data		Aggregated Data		Raw Data
More Dimension(s)	Last Dimension	More Measure(s)	Last Measure	
One Less Dimension	Measures Only on Source Data	One Less Measure	*COUNT* across All Dimensions	One Less Measure

Axis Brushing

The process of selecting multiple *discrete* values or a range of *continuous* values on the scale of a chart axis and producing another chart as a result is called **axis brushing**. If a single value needs to be selected, **axis drill-down** is used instead (*Cf.* Axis Drill-Down). A similar process exists for marks (*Cf.* Mark Brushing).

Selecting multiple *discrete* values or a range of *continuous* values for a variable is different from selecting a single *discrete* value of the variable, because it does not remove the variable from the dataset being visualized, unlike what is done for *axis drill-down*. Instead, it just applies a filter on the data being visualized by the chart. This fundamental difference explains why *axis brushing* and *axis drill-down* work in fundamentally different ways.

If the brushed axis is *independent*, its values are *discrete*, and the dataset being visualized is the result of a group-by aggregation (*pivot*). The selection of *discrete* values on this axis is equivalent to the application of a filtering facet related to the *independent* variable bound to the axis, either upstream or downstream of the pivot aggregation (both lead to the exact same results).

If the brushed axis is *dependent*, its values could be *discrete* or *continuous*, and it does not matter whether the dataset being visualized is raw data or aggregated data. In either case, the dataset at hand must be filtered directly, downstream of any transformation it could be the result of.

Since the brushing of *independent* and *dependent* axes can be handled in the same way through downstream filtering of the dataset at hand, they do not need to be distinguished, and the process of *axis brushing* turns out to be a much simpler one that the process of *axis drill-down*.

Chart Marks

According to the Vega Visualization Grammar, **marks** are the basic visual building block of a visualization [or chart]. Marks provide basic shapes whose properties can be set according to backing data. Mark properties can be simple constants or data fields, and **scales** can be used to map from data to property values. The following properties are shared by all marks[10]:

Property	**Type**	**Description**
x	Number	The first (typically left-most) x-coordinate.
x2	Number	The second (typically right-most) x-coordinate.
xc	Number	The center x-coordinate (incompatible with x and x2).
width	Number	The width of the mark (if supported).
y	Number	The first (typically top-most) y-coordinate.
y2	Number	The second (typically bottom-most) y-coordinate.
yc	Number	The center y-coordinate (incompatible with x and x2).
height	Number	The height of the mark (if supported).
opacity	Number	The overall opacity.
fill	Color	The fill color.
fillOpacity	Number	The fill opacity.
stroke	Color	The stroke color.
strokeWidth	Number	The stroke width, in pixels.
strokeOpacity	Number	The stroke opacity.

[10] Courtesy of the Vega Visualization Grammar. Some cosmetic properties have been omitted.

Mark Types

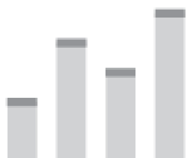
Much like the Vega Visualization Grammar, *Principia Pictura* relies on a basic set of marks. Nevertheless, more sophisticated marks could be conceived in order to produce specific visuals, without fundamentally altering the underlying grammar offered by the current framework.

- **Rectangles**
- **Areas**
- **Lines**
- **Paths**
- **Arcs**
- **Symbols**
- **Images**
- **Texts**

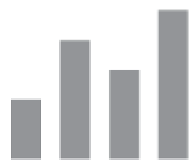
Rectangles

This mark does not have any additional properties.

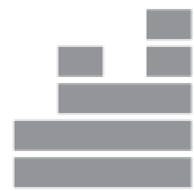
The **rectangle** is the most common mark. It is used by the following charts:



Level Chart



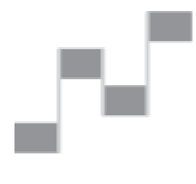
Bar Chart



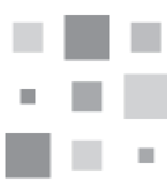
Castle Chart

Segment Chart

Block Chart

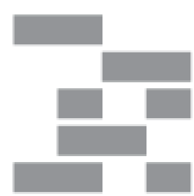


Band Chart



Heatmap Chart

Tape Chart



Gantt Chart

Areas

Property	**Type**	**Description**
orient	Category	The orientation of the area mark (horizontal or vertical).
interpolate	Category	The line interpolation method to use.
tension	Number	Depending on the interpolation type, sets the tension parameter.

Area Chart

Range Chart

Ribbon Chart

The following line interpolation methods can be used:

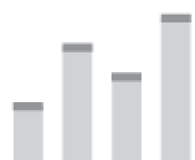
- **linear** (*Cf.* Linear Interpolation)
- **step-before** (alternate between vertical and horizontal segments, as in a step function)
- **step-after** (alternate between horizontal and vertical segments, as in a step function)
- **basis** (a B-spline; with control point duplication on the ends)
- **basis-open** (an open B-spline; may not intersect the start or end)
- **cardinal** (a Cardinal spline, with control point duplication on the ends)
- **cardinal-open** (an open Cardinal spline, may not intersect the start or end)
- **monotone** (monotone cubic interpolation that preserves monotonicity in *y*)

Lines

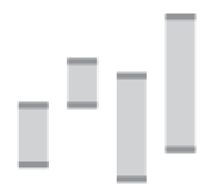
Property	**Type**	**Description**
interpolate	Category	The line interpolation method to use.
tension	Number	Depending on the interpolation type, sets the tension parameter.

Line Chart

Range Chart



Level Chart



Segment Chart

The line interpolation methods available are the same as the one defined for *area charts*.

It should be noted that some charts use several marks for depicting the same set of variables. For example, the *range chart* uses both *area* and *line* marks. Similarly, the *level chart* uses both *rectangle* and *line* marks. And the *segment chart* uses *line* marks for the *baseline* and *endline*, while using a *rectangle* mark for the difference between *endline* and *baseline* (*gain*).

Paths

Property	**Type**	**Description**
path	Geometry	A path definition (*e.g.* GeoJSON geometry).

The **path** mark is used for *geographic maps* like *choropleth maps*.

Arcs

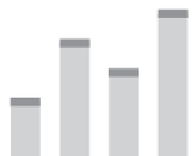
Property	Type	Description
innerRadius	Number	The inner radius of the arc, in pixels.
outerRadius	Number	The outer radius of the arc, in pixels.
startAngle	Number	The start angle of the arc, in radians.
endAngle	Number	The end angle of the arc, in radians.

Arc Chart

Symbols

Property	Type	Description
size	Number	The pixel area of the symbol.
shape	Category	The symbol shape to use.
		circle | *square* | *cross* | *diamond* | *triangle-up* | *triangle down*

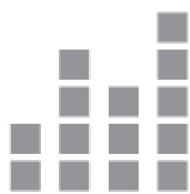
Line Chart



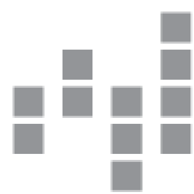
Level Chart

Segment Chart

Bit Chart

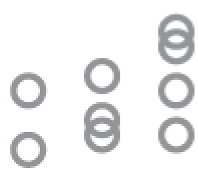


Tape Chart



Gantt Chart

Heatmap Chart



Circle Chart

Scatter Plot

Note: on the *line chart*, a *symbol* mark can be used to outline breakpoints on lines.

Images

Property	**Type**	**Description**
url	String	The URL from which to retrieve the image.
align	Category	The horizontal alignment of the image (left | right | center).
baseline	Category	The vertical alignment of the image (top | middle | bottom).

The **image** mark can be used for *geographic maps* like *choropleth maps* or *location maps*.

Mark Colors

All chart marks can be colored, and the selection of mark colors is a difficult exercise, for several important reasons. First, colors should not induce any unwelcome bias; second, certain combinations of colors can be difficult to distinguish for people affected with any of the many different forms of color blindness; third, there is no general consensus regarding the aesthetically pleasing nature of a certain combination of colors. In short, making colors look good while remaining legible is a substantial challenge.

In order to properly tackle this challenge, one should use a **color palette** that is particularly suited to the variable being depicted by the *color* axis. To do so, *Principia Pictura* recommends a set of *color palettes* that are optimized for the most common typologies of variables that can be depicted through a *color* axis. These palettes are inspired by Color Brewer 2.0, which was developed by Cynthia Brewer and Mark Harrower at The Pennsylvania State University.

Boolean Color Palette

A **boolean color palette** is made of two colors and is used for *boolean* variables. This typology is a particular case of the *nominal* typology, hence could use a *nominal color palette*. Nevertheless, it deserves its own palette, because many *boolean* variable are defined with sentiment metadata, indicating whether *better* is *TRUE* or *FALSE*. Taking advantage of such metadata, marks for *better* values could be depicted in *green*, while marks for *worse* values could be depicted in *red*. Granted, this color mapping is highly culturally dependent, and different mappings might be more appropriate within different cultural contexts.

Nominal Color Palette

A **nominal color palette** is made of up to 20 colors (usually) and is used for *nominal* variables. Colors in a good *nominal color palette* are highly differentiated from each other and can be distinguished by most people affected by some forms of color blindness (but not all).

Ordinal Color Palette

An **ordinal color palette** is made of up to *10* colors (usually) and is used for *ordinal* variables. Colors in an *ordinal color palette* are usually produced from a single hue with incremental levels of saturation, from low to high. It should be noted that some *ordinal* variables might require their *ordinal color palettes* to be reversed in order to better convey their meanings. For example, *priorities* with low ordinal values are usually more *urgent* than *priorities* with higher values, hence might be better depicted through colors with higher saturations.

Diverging Color Palette

A **diverging color palette** is made of up to 21 colors (*10 + 10 + 1*) and is used for *signed ordinal* variables. *Diverging color palettes* are usually produced by combining two *ordinal color palettes* and selecting a neutral color for the **zero reference color** (*Cf.* Signable Axis).

Gradient Color Spectrum

Finally, a **gradient color spectrum** can display a virtually unlimited number of colors and is used for *continuous* variables. Most *gradient color spectrums* are defined with one or two color hues (one hue and a variable saturation, or two extreme hues and a variable hue), but some might be defined with additional hues related to specific thresholds[11].

[11] Future revisions of *Principia Pictura* could specify *gradient color spectrum* in more details.

Mark Drill-Down

The process of selecting a mark on a chart and producing another chart as a result is called **mark drill-down**. If a set of marks needs to be selected, **mark brushing** must be used instead (*Cf.* Mark Brushing). A similar process exists for axes (*Cf.* Axis Drill-Down).

Mark drill-down works in the same way as *axis drill-down* does, with the main difference that multiple variables (either *independent* or *dependent*) can be removed at once. Therefore, the former can achieve in a single user interaction what the latter achieves through multiple ones.

Mark Brushing

The process of selecting several marks on a chart and producing another chart as a result is called **mark brushing**. If a single mark needs to be selected, **mark drill-down** is used instead (*Cf.* Mark Drill-Down). A similar process exists for axes (*Cf.* Axis Brushing).

Mark brushing works in the same way as *axis brushing* does, with the main difference that multiple variables (either *independent* or *dependent*) can be filtered at once. Therefore, the former can achieve in a single user interaction what the latter achieves through multiple ones.

Variable Binding

When binding a variable to an axis, the following conditions must be fulfilled:

- The variable's relation (*independent* or *dependent*) matches the axis' relation;
- The variable's typology matches the axis' typology or is more refined;
- The axis is signable if the variable is signed;
- The axis is not yet bound to another variable (unless it support *multiplicity*).

Chart Classification

Charts defined by *Principia Pictura* are organized in four main classes:

- **Cartesian Charts** — charts with a *continuous vertical* axis.
- **Tabular Charts** — charts with a *discrete vertical* axis.
- **Geographic Maps** — visuals produced with a geographic map as background.
- **Other Visuals** — any visual which does not fit into any of the classes defined above.

The following letters are used to describe the properties of charts and their axes:

- `Δ` — *Differential*
- `I` — *Independent*
- `D` — *Dependent*
- `S` — *Signable*
- `*` — *Required*
- `1` — One required and only one
- `n` — One required at least
- `M` — *Multiple*
- `C` — *Cohort*
- `d` — *Details*

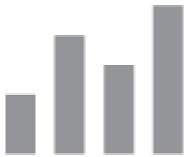
The polarization column provides the polar alias of every axis (*Cf.* Chart Polarization).

Cartesian Charts

Cartesian charts are defined with a *continuous vertical* axis, and most are quite common.

- **Bar Chart**
- **Level Chart**
- **Area Chart**
- **Line Chart**
- **Block Chart**
- **Segment Chart**
- **Ribbon Chart**
- **Range Chart**
- **Circle Chart**
- **Scatter Plot**

Bar Chart



Axis	Polarization	Typology	Properties
Horizontal	Sector	Discrete	I S * C
Vertical	Radius	Extensive	D S 1 M d
Width	Angle	Absolutizable	D 1
Color	Color	Discretizable	D S
	Polar Chart:	*Polar Bar Chart*	

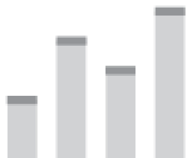
Horizontal Incremental Counterpart: *Area Chart*

Vertical Intensive Counterpart: *Level Chart*

Tabular Counterpart: *Dot Chart*

Note: The *width* axis is essential to its polar counterpart (*angle* in the *polar bar chart*).

Level Chart



Axis	Polarization	Typology	Properties
Horizontal	Sector	Discrete	I S * C
Vertical	Radius	Intensive	D S * M
Size	Size	Absolutizable	D
Value	Value	Discretizable	D
Symbol	Symbol	Discrete	D
	Polar Chart:	*Polar Level Chart*	

Horizontal Incremental Counterpart: *Line Chart*

Vertical Extensive Counterpart: *Bar Chart*

Tabular Counterpart: *Bit Chart*

Extension: The *level chart* could be extended with a *color* axis.

Area Chart

Axis	Polarization	Typology	Properties	
Horizontal	Sector	Incremental	I S *	
Depth	Layer	Lexical	I S	
Vertical	Radius	Extensive	D S *	d
Color	Color	Discretizable	D S	
Symbol	Symbol	Discrete	D	
	Polar Chart:	*Polar Area Chart*		

Horizontal Nominal Counterpart: *Bar Chart*

Vertical Intensive Counterpart: *Line Chart*

Tabular Counterpart: *Castle Chart*

Usage: The *depth* axis is used to produce an extensive counterpart to the joy division chart.

Note: The *symbol* axis is used to outline breakpoints on lines outlining the areas.

Line Chart

Axis	Polarization	Typology	Properties
Horizontal	Sector	Incremental	I S *
Depth	Layer	Lexical	I S
Vertical	Radius	Intensive	D S * M d
Color	Color	Discretizable	D S
Symbol	Symbol	Discrete	D
	Polar Chart:	*Polar Line Chart*	

Horizontal Nominal Counterpart: *Level Chart*

Vertical Extensive Counterpart: *Area Chart*

Tabular Counterpart: *Band Chart*

Usage: The *depth* axis is used to produce a joy division chart.

Note: The *symbol* axis is used to outline breakpoints on lines.

Block Chart

Axis	Polarization	Typology	Properties
Horizontal	Sector	Discrete	I S * C
Baseline	Baseline	Extensive	D S *
Endline	Endline	Extensive	D S 1
Gain	Gain	Extensive	D S 1 d
Δ	Polar Chart:	*Polar Block Chart*	

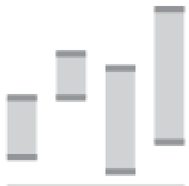
Horizontal Incremental Counterpart: *Ribbon Chart*

Vertical Intensive Counterpart: *Segment Chart*

Tabular Counterpart: *Train Chart*

Extension: The *block chart* could be extended with *color* and *width* axes.

Segment Chart



Axis	Polarization	Typology	Properties
Horizontal	Sector	Discrete	I S * C
Baseline	Baseline	Intensive	D S *
Endline	Endline	Intensive	D S 1
Gain	Gain	Extensive	D S 1 d
Δ	Polar Chart:	*Polar Segment Chart*	

Horizontal Incremental Counterpart: *Range Chart*

Vertical Extensive Counterpart: *Block Chart*

Extension: The *segment chart* could be extended with *color* and *width* axes.

Ribbon Chart

Axis	Polarization	Typology	Properties	
Horizontal	Sector	Incremental	I S *	
Baseline	Baseline	Extensive	D S *	
Endline	Endline	Extensive	D S 1	
Gain	Gain	Extensive	D S 1	d
Δ	Polar Chart:	*Polar Ribbon Chart*		

Horizontal Nominal Counterpart: *Block Chart*

Vertical Intensive Counterpart: *Range Chart*

Tabular Counterpart: *Tape Chart*

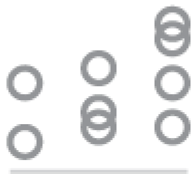
Range Chart

Axis	Polarization	Typology	Properties	
Horizontal	Sector	Incremental	I S *	
Baseline	Baseline	Intensive	D S *	
Endline	Endline	Intensive	D S 1	
Gain	Gain	Extensive	D S 1	d
Δ	Polar Chart:	*Polar Range Chart*		

Horizontal Nominal Counterpart: *Segment Chart*

Vertical Extensive Counterpart: *Ribbon Chart*

Circle Chart



	Axis	Polarization	Typology	Properties	
	Horizontal	Sector	Discrete	D S *	d
	Vertical	Radius	Intensive	D S *	
	Size	Size	Absolutizable	D	
		Polar Chart:	*Polar Circle Chart*		

Horizontal Continuous Counterpart: *Scatter Plot*

Note: The *circle chart* has no *independent* axis.

Scatter Plot

Axis	Polarization	Typology	Properties
Horizontal	Sector	Continuous	D S *
Vertical	Radius	Continuous	D S *
Angle	Angle	Sectorizable	D S
Length	Length	Quantizable	D
Size	Size	Quantizable	D
Color	Color	Discretizable	D S
Symbol	Symbol	Discrete	D
Value	Value	Discretizable	D

Polar Chart: *Polar Scatter Plot*

Horizontal Discrete Counterpart: *Circle Chart*

Geographic Counterpart: *Location Map*

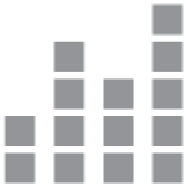
Note: The *scatter plot* has no *independent* axis.

Tabular Charts

Tabular charts are defined with a *discrete vertical* axis, and many are quite unusual.

- **Dot Chart**
- **Bit Chart**
- **Castle Chart**
- **Band Chart**
- **Train Chart**
- **Tape Chart**
- **Gantt Chart**
- **Heatmap Chart**
- **Table Chart**

Dot Chart

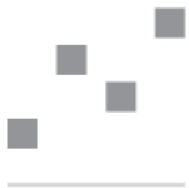


Axis	Polarization	Typology	Properties
Column	Sector	Discrete	I S * C
Row	Track	Cardinal	D S * d
	Polar Chart:	*Polar Dot Chart*	

Horizontal Incremental Counterpart: *Castle Chart*

Cartesian Counterpart: *Bar Chart*

Bit Chart

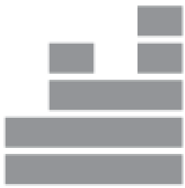


Axis	Polarization	Typology	Properties
Column	Sector	Discrete	I S *
Row	Track	Discrete	D S *
Size	Size	Absolutizable	D
Color	Color	Discretizable	D S
Symbol	Symbol	Discrete	D
	Polar Chart:	*Polar Bit Chart*	

Horizontal Incremental Counterpart: *Band Chart*

Cartesian Counterpart: *Level Chart*

Castle Chart

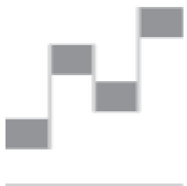


Axis	Polarization	Typology	Properties
Column	Sector	Incremental	I S *
Row	Track	Ordinal	D S *
	Polar Chart:	*Polar Castle Chart*	

Horizontal Nominal Counterpart: *Dot Chart*

Cartesian Counterpart: *Area Chart*

Band Chart

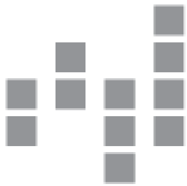


	Axis	Polarization	Typology	Properties
	Column	Sector	Incremental	I S *
	Row	Track	Ordinal	D S *
		Polar Chart:	*Polar Band Chart*	

Horizontal Nominal Counterpart: *Bit Chart*

Cartesian Counterpart: *Line Chart*

Train Chart



Axis	Polarization	Typology	Properties
Horizontal	Sector	Discrete	I S * C
Baseline	Baseline	Ordinal	D S *
Endline	Endline	Ordinal	D S 1
Gain	Gain	Cardinal	D S 1 d
Δ	Polar Chart:	*Polar Train Chart*	

Horizontal Incremental Counterpart: *Tape Chart*

Cartesian Counterpart: *Block Chart*

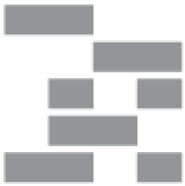
Tape Chart

Axis	Polarization	Typology	Properties
Horizontal	Sector	Incremental	I S *
Baseline	Baseline	Ordinal	D S *
Endline	Endline	Ordinal	D S 1
Gain	Gain	Cardinal	D S 1
Δ	Polar Chart:	*Polar Tape Chart*	

Horizontal Nominal Counterpart: *Train Chart*

Cartesian Counterpart: *Ribbon Chart*

Gantt Chart



Axis	Polarization	Typology	Properties
Start	Start	Incremental	D S *
Duration	Duration	Cardinal	D *
Completion	Completion	Absolute	D
Parent	Parent	Hierarchical	D
Predecessor	Predecessor	Hierarchical	D
Color	Color	Discretizable	D S
	Polar Chart:	*Polar Gantt Chart*	

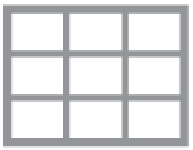
Note: The *gantt chart* has no *independent* axis.

Heatmap Chart

Axis	Polarization	Typology	Properties
Column	Sector	Discrete	I *
Row	Track	Discrete	I *
Color	Color	Discretizable	D S n
Height	Radius	Absolutizable	D n
Width	Angle	Absolutizable	D n
Symbol	Symbol	Discrete	D n
	Polar Chart:	*Polar Heatmap Chart*	

Note: The *heatmap chart* is one of the most flexible *tabular charts*.

Table Chart



Axis	Polarization	Typology	Properties
Column	Sector	Discrete	I * M
Row	Track	Discrete	I * M
Values	Values	Discretizable	D n M
Size	Size	Absolutizable	D n
Color	Color	Discretizable	D S n
Symbol	Symbol	Discrete	D n
	Polar Chart:	*Polar Table Chart*	

Note: The *table chart* is the only chart that supports multiple *multiplicity axes*.

Geographics Maps

Geographic maps are produced with a geographic map as background.

- **Choropleth Map**
- **Location Map**

Choropleth Map

Axis	Polarization	Typology	Properties
Region	N/A	Geometrical	I *
Size	N/A	Absolutizable	D
Color	N/A	Quantizable	D S

Warning: The chart thumbnail shown above is incorrect and will be updated soon.

Note: The *size* axis can be used to scale the *region's* geometry.

Location Map

Axis	Polarization	Typology	Properties
Longitude	N/A	Longitudinal	D S *
Latitude	N/A	Longitudinal	D S *
Angle	N/A	Angular	D S
Length	N/A	Quantizable	D
Size	N/A	Absolutizable	D
Color	N/A	Discretizable	D S
Symbol	N/A	Discrete	D
Value	N/A	Discretizable	D

Cartesian Counterpart: *Scatter Plot*

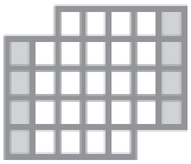
Note: The *location map* has no *independent* axis.

Other Visuals

Other visuals are visuals that do not fit into any standard class.

- **Calendar**
- **Arc Plot**
- **Parallel Coordinates**
- **Contour Plot**
- **Surface Plot**

Calendar



	Axis	Polarization	Typology	Properties
	Period	N/A	Calendar	I *
	Color	N/A	Quantizable	D S *

Note: The *calendar* visual must be localized.

Arc Plot

Axis	Polarization	Typology	Properties
Source	Source	Hierarchical	I *
Target	Target	Hierarchical	I *
Start	Start	Intensive	D S
End	End	Intensive	D S
Offset	Offset	Absolutizable	D S
Color	Color	Discretizable	D S
	Polar Chart:	*Chord Diagram*	

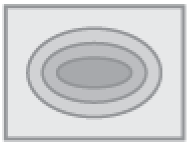
Note: For 3 *source/target* bases, the *hive plot* can also be used as polar counterpart to the *arc plot*.

Parallel Coordinates

Axis	Polarization	Typology	Properties
Coordinates	Coordinates	Any	D M *
	Polar Chart:	*Star Plot*	

Note: *Parallel coordinates* have no *independent* axis.

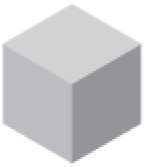
Contour Plot



Axis	Polarization	Typology	Properties
X (Abscissa)	N/A	Incremental	I S *
Y (Ordinate)	N/A	Incremental	I S *
Z (Applicate)	N/A	Quantizable	D S *

Three-Dimensional Counterpart: *Surface Plot*

Surface Plot



Axis	Polarization	Typology	Properties
X (Abscissa)	N/A	Incremental	I S *
Y (Ordinate)	N/A	Incremental	I S *
Z (Applicate)	N/A	Quantizable	D S *
Color	N/A	Quantizable	D S

Two-Dimensional Counterpart: *Contour Plot*

Chart Recommendations

It is beyond the scope of *Principia Pictura* to define precise algorithms for the recommendation of the most suitable chart in relation to a given dataset. Nevertheless, some basic heuristics can be outlined in order to guide anyone interested in implementing a *chart rendering engine*. Furthermore, the user of such a tool will certainly benefit from understanding the rules that guide its behavior when charts are automatically produced.

The problem at hand is one of *matching*: on one side, a dataset made of variables defined with their relations (*independent* or *dependent*), typologies, and properties (*e.g. signed*). On the other side, a collection of visuals made of axes defined with their relations, typologies, and properties as well. In the middle, a basic set of variable binding rules (*Cf.* Variable Binding). From there, a set of valid visuals emerges, and they must be ranked in order for a single recommendation to be made. And if no basic visual comes out as a match, more advanced ones can be suggested by taking advantage of some visual transformation techniques (*Cf.* Visual Transformations).

In order for visuals to be ranked, a score must be computed for each and every visual. In this particular context, a visual is defined as an individual binding of the datasets variables against a chart's axes. With such a definition, multiple valid visuals can be produced for the same dataset using the same chart, by simply mapping different variables of the same dataset to different axes of the same chart. Therefore, the ranking applies to specific visuals, not generic charts.

In order to rank multiple visuals produced from the same charts, the axes of a chart themselves can be ranked, by taking two factors into consideration: first, the *hierarchy of cognitive processes* defined by William Cleveland and Robert McGill (*Cf.* Cognitive Context); second, the degree of familiarity of certain axes in relation to other (*Cf.* Cultural Context) — for example, with the *bar chart*, the *vertical* axis is much more common than the *width* axis.

The absolute ranking of axes can also be complemented by a relative ranking, taking several additional factors into consideration. For example, the binding of a variable onto a first-class axis should be prioritized over the use of *axis multiplicity*, *details*, or *cohorts*. Similarly, some axes are legible only when bound to low cardinality variables, therefore some variable/axis combinations will work better than others in relation to a variable's cardinality and should be scored accordingly by the algorithm.

Once all possible visuals for all compatible charts have been scored individually, they must be ranked in relation to each other. In order to support this process, some additional parameters can be introduced. For example, charts can be given an absolute priority. This priority allows a *contour plot* to be prioritized over a *surface plot*, because the former is usually more legible than the latter, since it does not rely on any perspective projection.

Finally, some charts are intrinsically better than others at visualizing certain combinations of variables, or certain types of variables. A very typical example is the family of *differential charts*, (*Cf.* Differential Secondary Charts), which are particularly well suited for the visualization of differential variables (*baseline* and *endline* or *baseline* and *gain*) — if a non-differential chart were to be used for visualizing a differential dataset, the result would most likely be catastrophic.

This very basic set of rules is just an example of what a *chart rendering engine* could do to automate the recommendation of suitable charts for an arbitrary dataset. More advanced rules can certainly be developed, by taking advantage of additional aspects such as contextual priorities, statistical analysis, visual complexity, and many others. Extending this set of heuristics is certainly one of the main priorities for future revisions of the present work.

Visual Transformations

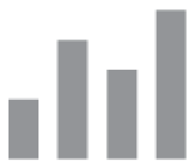
The basic set of charts proposed by *Principia Pictura* (*Cf.* Chart Classification) can be conceived of as a collection of LEGO® pieces that need to be assembled into sets in order to produce some interesting visuals. And even some very common charts like *pie chart* or *bar and line chart* cannot be produced without transformation. This is by design, for it allows *Principia Pictura* to remain very generic and flexible, while conferring it the ability to produce virtually any visual.

Some transformations are simple and cosmetic (*Cf.* Chart Orientation), others are complex and transformative (*Cf.* Mark Subdivision), and one is both cosmetic and transformative (*Cf.* Chart Polarization). Developing some degree of familiarity with these transformation (especially the simpler ones) is essential for getting the benefits offered by *Principia Pictura*.

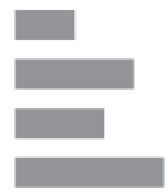
- **Chart Orientation** (*Cf.* Chart Orientation)
- **Chart Tabulation** (*Cf.* Chart Tabulation)
- **Chart Polarization** (*Cf.* Chart Polarization)
- **Chart Composition** (*Cf.* Chart Composition)
- **Chart Tiling** (*Cf.* Chart Tiling)
- **Mark Subdivision** (*Cf.* Mark Subdivision)

Chart Orientation

The **orientation** of a chart is controlled by swapping its *horizontal* and *vertical* axes. It is supported as a transformation instead of a dedicated set of basic charts, because it applies to all *cartesian* and *tabular* charts. This helps keep the set of standard charts as small as possible.



Standard Bar Chart



Rotated Bar Chart

Several factors can drive the *orientation* of a chart:

- Overall legibility;
- Amount of real estate available for its rendering;
- Conventions or cultural norms;
- Aesthetic preferences and other cosmetic factors.

Nevertheless, data analysts should always attempt to keep the most relevant *dimension* on the *horizontal* axis and the most relevant *measure* on the *vertical* axis. Also, changing the *orientation* of a chart is not always limited to swapping its *horizontal* and *vertical* axes. Sometimes, it also involves reversing the order of one or two of the variables bound to these axes. For example, if an *epochal* variable is bound to the *horizontal* axis going from left to right, the swapping of the *horizontal* and *vertical* axes will depict the *epochal* variable against the vertical axis going from bottom to top, but convention usually demands the opposite direction. This convention probably originates from the fact that a page is usually read from top to bottom. Also, most Western writing systems usually go from left to right. Therefore, time usually go from left to right and from top to bottom (unlike most *numerical* variables, which grow from bottom to top).

Chart Tabulation

A popular technique for increasing the **dimensionality** of a visual (its number of *dimensions*) is called **chart tabulation**, and is also known as small multiples. A *small multiple* (sometimes called *trellis chart*, *lattice chart*, grid chart, or *panel chart*) is a series of similar charts using the same scale and axes, allowing them to be easily compared. It uses multiple views to show different partitions of a dataset. The term was popularized by Edward Tufte. [Source: Wikipedia]

Chart tabulation can be used to add one or two dimensions (*independent* variables) to a visual, either horizontally, vertically, or both. A two-dimensional array of small multiples could be bound to one or two dimensions, and could easily display up to *100* (*10 × 10*) individual charts.

Chart tabulation is an attractive option when the most suitable basic chart available to visualize a certain dataset has too many dimensions to remain legible. *Chart tabulation* is also a very effective way of comparing individual values of an *independent* variable or pairs of values of two *independent* variables, without the visual clutter usually created by high-dimensionality charts.

Nevertheless, *chart tabulation* should only be used when tabulating simple charts with low dimensionality and low visual density. Otherwise, the resulting visual will have a very high dimensionality and a very high visual density, making it very difficult to interpret.

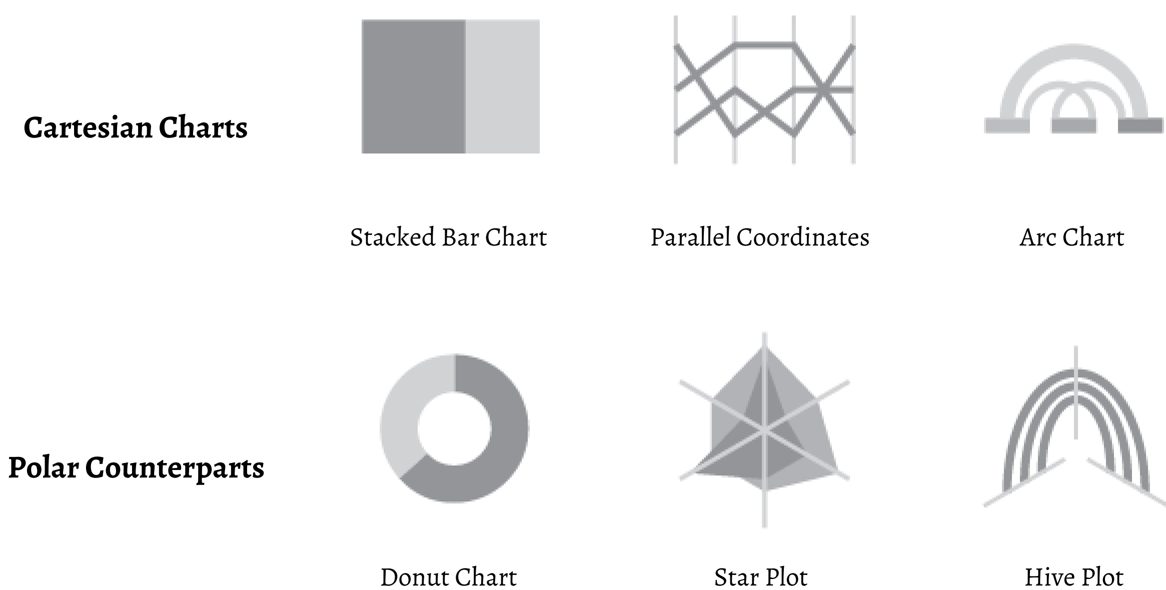
When designing a visual, always attempt to convey meaning instead of exhibiting mastery.

Chart Polarization

Out of the box, *Principia Pictura* cannot produce one of the simplest charts: the *pie chart*.

How could this be?

The answer to that question is quite simple: the basic set of charts proposed by *Principia Pictura* are all *cartesian* or *tabular* (*discrete* counterpart to *cartesian*), at the exception of *geographic maps* and a handful of highly specific visuals. As a result, they all use a cartesian coordinate system. Unfortunately, charts like *pie chart*, *star plot*, or *hive plot* use a polar coordinate system.



An obvious solution to this problem is to support both coordinate systems and to extend the basic set of charts proposed by *Principia Pictura* (*Cf.* Chart Classification). But there is a more elegant and much more powerful solution: it consists in developing a technique of *polarization* capable of transforming any *cartesian* chart into its *polar* counterpart.

This solution is made possible by the fact that most (if not all) conventional *polar* charts have conventional *cartesian* counterparts that are already offered by *Principia Pictura*. As a result, all that is needed is some additional metadata (like *polar aliases* for the names of chart axes) and a (now very powerful) *chart rendering engine* capable of establishing a visual bijection between the two coordinate systems (this is obviously easier written than done).

Polarization is a very powerful technique for two main reasons:

First, it can be used to produce a wide range of *polar* charts that so far have been ignored or underutilized by the communities of statisticians and data analysts, either by simple ignorance or because of the lack of suitable tools that could produce such charts from existing datasets.

Second, it can be used to produce an amazingly powerful family of charts called circos charts, which were invented by Martin Krzywinski (BC Genome Sciences Centre). Martin Krzywinski, alongside Mike Bostock and Jeffrey Heer, is one of the genius pioneers of modern visualization. Mike Bostock established with D3.js the benefits of a declarative approach for visualization, while Jeffrey Heer and his team (of which Mike Bostock used to be a member) created the first visualization grammar with Vega.

It is the opinion of this author that Martin Krzywinski's major contributions to the field of visualization are twofold: first, establishing the benefits of the circular layout (*polarization*) for dealing with high-dimensionality datasets; second, reaffirming the mandate of quantitative aggregation when visualizing graph datasets (*Cf.* Hive Plots). These two avenues of research will most certainly drive unprecedented progress in the field, especially when combined together for the visualization of high-cardinality and high-dimensionality graph-oriented datasets.

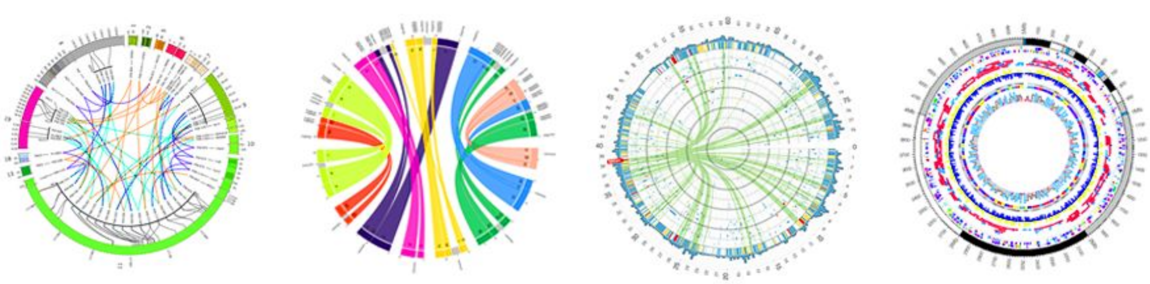
Predictions aside, the following excerpts from the Circos website explain why *circos* work.

Circos charts use a circular composition to show connections between objects or between positions, which are difficult to visually organize when the underlying layout is linear (or a graph, which can quickly become a hairball). In many cases, a linear layout makes impossible keeping the relationship lines from crossing other structures, [thereby deteriorating] the effectiveness of the graphic.

In addition to its strength in depicting relationships, the circular form itself has a number of useful properties which are not shared by a rectilinear layout. The circular form encourages eye movement to proceed along curved lines, rather than in a zigzag fashion in a square or rectangular figure.

Within the circle, resolution varies linearly and increases with radial position. This makes the center of the circle ideal for compactly displaying summary statistics or indicating points of interest (*i.e.* low resolution data) which the reader can then follow outward to explore the data in greater detail (*i.e.* high resolution data).

Also, for a given square area on a page (*e.g.* square of side x), the circular layout can support a larger data domain (*i.e.* $2\pi x \sim 6x$) than the square (*i.e.* x). More data can be shown within a given space on a page (or within narrower field of vision).



Circos Charts, Courtesy of Martin Krzywinski

Chart Composition

Another technique for increasing the *dimensionality* of a visual is called **chart composition**, and it consists in displaying multiple charts next to each other (*Cf.* Chart Juxtaposition), or on top of each other (*Cf.* Chart Superimposition).

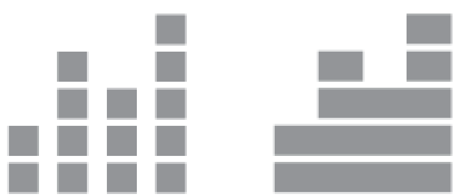
These two techniques complement the chart polarization technique (*Cf.* Chart Polarization), in the sense that a typical *circos chart* is actually produced by the *polarization* of a set of charts *composed* through *juxtaposition* and *superimposition*.

In other words, the collection of compatible visual transformations offered by *Principia Pictura* (*Cf.* Visual Transformations) is akin to the collection of assembly techniques offered by LEGO® pieces (stud of course, but also technic pin 3673, and many others): they can be used together to design very sophisticated assemblies from fairly simple primitives.

Chart Juxtaposition

Chart juxtaposition is a type of *chart composition* consisting in juxtaposing different charts next to each other, either *side by side*, *top to bottom*, or both. When doing so, axes that are parallel to each other must be *congruent* with each other (same typology, unit of measurement, and scale).

Side by Side Juxtaposition



Top to Bottom Juxtaposition

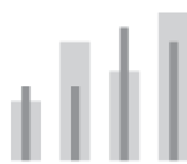
Chart juxtaposition should not be confused with *chart tabulation* (*Cf.* Chart Tabulation). Indeed, the former juxtaposes and aligns different charts next to each other, while the latter replicates the same chart multiple times in relation to one or two additional *dimensions*.

Chart juxtaposition should be used when a small number of relatively simple charts (2, 3, or 4) can be used to visualize a dataset that cannot be suitably visualized through a single chart. Also, by splitting the dataset into multiple sub-datasets, *chart juxtaposition* makes it easier to emphasize the interplay of certain subsets of dimensions and measures, while keeping all variables into an integrated context (a single visualization). Nevertheless, taking proper advantage of this technique requires a solid understanding of both *Principia Data* and *Principia Pictura*, and a profound level of intimacy with the dataset at hand.

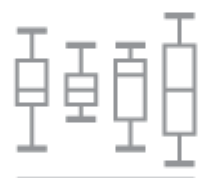
Chart Superimposition

Chart superimposition is a type of *chart composition* consisting in superimposing different charts on top of each other. When doing so, all *horizontal* axes of the superimposed charts must be *congruent* with each other (same typology, unit of measurement, and scale), or multiple parallel scales must be defined, and the same applies to *vertical* axes. This is how *Principia Pictura* can produce charts such as the *bar and line chart*, the *bullet chart*, or the *box chart*.

Bar and Line Chart



Bullet Chart



Box Chart

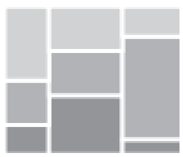
Chart superimposition is a delicate exercise, even more so than *chart juxtaposition*. Therefore, it is usually reserved to the *a priori* construction of charting templates that can be used to produce conventional charts from arbitrary datasets *a posteriori*. Nevertheless, judicious application of this technique can be highly effective when trying to exhibit some correlations.

Chart Tiling

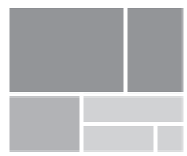
Chart tiling is the simple process of displaying multiple charts on a single canvas, in relation to a single dataset. Unlike *chart juxtaposition*, *chart tiling* does not require any axis alignment. As a result, the relative placement of charts on the canvas is free of any constraints. For example, the 4-Plot used for exploratory data analysis can be produced with chart tiling.

Mark Subdivision

One last technique for increasing the *dimensionality* of a visual is called **mark subdivision**, and it consists in subdividing a *rectangle* mark into smaller *rectangles*, across one additional *dimension*, or sometime two or more. This technique is possible with at least three types of subdivision:



Mosaic Subdivision

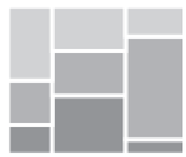


Treemap Subdivision

Partition Subdivision

Mosaic Subdivision

The **mosaic subdivision** consists in drawing a *mosaic plot* (*Cf.* Two Dimensions) within the space occupied by the subdivided *rectangle* mark. This type of subdivision can be made across one, two, or an unlimited number of dimensions, even though two is a practical maximum.



Mosaic Subdivision

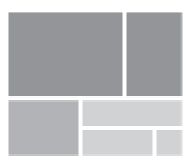
A typical application of this technique is when producing a *mosaic bar chart*, which is a *bar chart* for which every *bar* is its own *mosaic plot*. This allows three independent variables to be depicted through a single chart. Nevertheless, such a technique usually leads to a significant amount of visual clutter, which might go against the intended purpose.

Treemap Subdivision

Treemaps display hierarchical data as a set of nested rectangles. Each branch of the tree is given a rectangle, which is then tiled with smaller rectangles representing sub-branches. A leaf node's rectangle has an area proportional to a specified measure. Often the leaf nodes are colored to show a separate dimension of the data.

When the variables bound to the color and size axes are correlated in some way with the tree structure, one can often easily see patterns that would be difficult to spot in other ways, such as if a certain color is particularly relevant. A second advantage of treemaps is that, by construction, they make efficient use of space. As a result, they can legibly display thousands of items on the screen simultaneously. [Source: Wikipedia, with minor adjustments]

The **treemap subdivision** consists in drawing a *treemap* within the space occupied by the subdivided *rectangle* mark. As a result, it adds two dimensions to the chart which rectangle marks are being subdivided: a *hierarchical* one and a nominal one (bound to the new *color* axis).



Treemap Subdivision

By design, a *treemap* can depict a *hierarchical* dimension (*parent*) and a *nominal* one (*color*), but it can be used for two *nominal* dimensions as well, replacing the *hierarchical* dimension by a simpler *nominal* one. As such, the *treemap subdivision* can be considered as an alternative to the *mosaic subdivision*. When considering the two for a particular visualization, one might keep in mind that *treemaps* can usually accommodate a much larger number of items than *mosaic plots*. Nevertheless, this tolerance to high cardinality comes at the expense of mixing the values of the two dimensions across the chart, unlike the *mosaic plot*, which separates the two neatly.

Partition Subdivision

The last subdivision is the **partition subdivision**. Much like the *treemap subdivision*, it adds a *hierarchical* dimension (*parent*) and a *nominal* one (*color*), but it is much less space efficient. Nevertheless, what it loses in space efficiency, it gains in hierarchical legibility.

Partition Subdivision

It should be noted that a sunburst is a simple *polarization* of a *partition chart*.

Visual Enhancements

Visual enhancements encompass a very broad range of visual decorations that can be added to a chart in order to improve its ability to convey some important information. By definition, *visual enhancements* do not make any alterations to the dataset visualized by a chart, nor to the way the dataset's variables are bound to the chart's axes.

Nevertheless, they do not serve a purely cosmetic purpose. Instead, they have meaning, and their metadata definition can sometime lead to actual transformations (*e.g.* filters, pivots, *etc.*) during the course of a data exploration exercise.

- **Groups**
- **Clusters**
- **Highlights**
- **Differences**
- **Growths**
- **Trends**
- **References**
- **Labels**
- **Summaries**
- **Texts**
- **Breaks**
- **Connectors**
- **Paddings**

Groups

A **group** is a set or range of values for a given variable. A group can be visually depicted on the scale of the axis to which the variable is bound. Its purpose is to emphasize a set or range of values, inviting the analyst to directly perform an *axis brushing* on them (*Cf.* Axis Brushing).

Additionally, if all possible values for an axis are classified through *groups*, the set of *groups* can be used as a new *independent* variable to perform a group-by aggregation (*pivot*).

Finally, *groups* can be used to organize the values of *nominal* variables that have a medium cardinality — too many to be considered at once, too few to justify the promotion to a relation.

Clusters

A **cluster** is a set of marks directly related to data items (rows in the dataset). *Clusters* can be visually depicted through circles, rectangles, areas, paths, or colors, depending on the marks and colors already used by the chart. *Visual clusters* can be used to depict *statistical clusters* that are extracted through cluster analysis, or *ad hoc clusters* that are manually outlined.

The clustering of all data items across a set of *clusters* leads to the definition and population of an implicit nominal variable that can be used for analysis purpose (*e.g.* pivot dimension), and a proper user interface could facilitate the creation of this variable.

The definition of algorithms for *cluster analysis* is beyond the scope of *Principia Pictura*.

Highlights

A **highlight** is the visual highlighting of a mark or set of marks directly related to data items (rows in the dataset). *Highlights* can be depicted through callouts attached to a single mark or a set of marks. Their purpose is to outline outliers or important data items.

The definition of algorithms for *outlier detection* is beyond the scope of *Principia Pictura*.

Differences

A **difference** is the difference between two values of the same variable. For example, it could be the *vertical* difference between two *bars* on a *bar chart*. Typically, *differences* are displayed for two values depicted by two marks that are close to each other, but this does not have to be the case. Also, differences can be displayed for specific pairs of values, or for all consecutive pairs of values across a sorted axis (*lexical* typology and beyond).

Differences can be used to invite the analyst to transform a regular dataset into a differential one. (*Cf.* Differential Secondary Chart). This can be especially valuable for datasets that are defined with multiple sorted *independent* variables, and for which the *a priori* selection of the primary *differential dimension* (sorted *independent* variable against which differences are computed) is not entirely obvious to the analyst.

Differences are similar to *growths*, but the latter show *relative growths* instead of *absolute differences*.

Growths

A **growth** is the relative growth between two values of the same variable. For example, it could be the *vertical* growth between two *bars* on a *bar chart*. Typically, *growths* are displayed for two values depicted by two marks that are close to each other, but this does not have to be the case. Also, growths can be displayed for specific pairs of values, or for all consecutive pairs of values across a sorted axis (*lexical* typology and beyond).

Growths can be used to invite the analyst to transform a regular dataset into a differential one. (*Cf.* Differential Secondary Chart). This can be especially valuable for datasets that are defined with multiple sorted *independent* variables, and for which the *a priori* selection of the primary *differential dimension* (sorted *independent* variable against which differences are computed) is not entirely obvious to the analyst.

Growths are similar to *differences*, but the latter show *absolute differences* instead of *relative growths*.

Trends

A **trend** is a line or curve used to exhibit a relationship between a *dependent* variable and one or more *independent* variables. The *trend* can be defined with a common regression model, or a custom one. In addition, it can be complemented by a **confidence band** depicted as an area. Furthermore, a separate trend line could be displayed for every value of another low-cardinality *independent* variable. Common regression models include:

- Linear
- Logarithmic
- Exponential
- Polynomial

References

A **reference** is a single value outlined on an axis. While a reference is similar to a *group*, its depiction on a chart is similar to the one of a *trend*. The purpose of a *reference* and associated **reference band** (for a *discrete* axis) or **reference line** (for a *continuous* axis) is to outline thresholds and the position of data items in relation to them.

A *reference* could invite the analyst to perform an *axis drill-down* (*Cf.* Axis Drill-Down).

Labels

A **label** is a textual depiction of a variable's value alongside the mark used to depict the value. When using modern software applications for rendering charts, labels are usually displayed on demand in order to reduce visual clutter as much as possible. Nevertheless, some scenarios benefit from the permanent display of labels, either for all data items in relation to a particular axis, or for specific data items in relation to a particular *highlight* (*Cf.* Highlights).

When a *label* is displayed for a particular *measure* (*dependent* variable), its related *dimensions* (*independent* variables) can be displayed as well in order to provide some additional context for the *measure's* value. This context can be further extended by combining the display of labels with the display of *summaries* (*Cf.* Summaries).

Summaries

A **summary** is a partial aggregation related to a subset of data items on a chart. For example, subtotals could be displayed on top of every *stack* on a *stacked bar chart*. *Summaries* can also be displayed for individual data items. For example, the ratio of an individual *bar* within a *stack*.

Texts

A **text** is a static textual content added to a chart. Examples of *text* include:

- **Title**
- **Subtitle**
- **Description**
- **Explanation**
- **Link to original data source**
- **Link to chart documentation**
- **Copyright notice**
- **Credits**

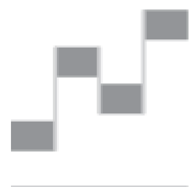
Breaks

A **break** is a point of discontinuity added to a sorted axis. A *break* is used when the presence of outliers skews the dataset in such a way that most data items become invisible or barely legible. A *break* defined against an axis is usually reflected on the marks used to depict the outliers and must be prominently featured so as not to mislead any unsuspecting data analyst.

A *break* can also be used to collapse ranges of values within which no data items are available.

Connectors

A **connector** is a *line* or set of *lines* used to visually connect multiple marks in an incremental manner. For example, connectors are used on a *band chart* to visually connect *rectangle* marks alongside the horizontal axis, thereby emphasizing the *incremental* nature of the latter.

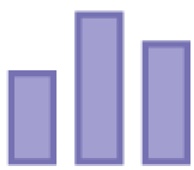


Band Chart

Paddings

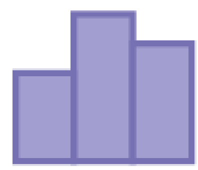
A **padding** is an horizontal or vertical spacing between marks. Some charts like *frequency chart* require padding, while other like *histogram* prevent it. For the ones that allow it, proper *padding* improves the legibility of charts, much like proper kerning improves the legibility of text. Unfortunately, few quantitative guidelines have been developed to date[12].

Padding



Frequency Chart

No Padding



Histogram

[12] Such guidelines could be the subject of a future revision to *Principia Pictura*.

Missing Data

Most data analysts use charts to visualize data, but few take advantage of them to visualize the absence of data. Somehow, statisticians have learned how to deal with missing data for quite a while, as witnessed by the number of options that can be used for treating missing values when computing summary statistics like averages. Unfortunately, this knowledge has yet to be transferred to the field of data visualization.

Principia Data advocates a systematic approach for the visualization of missing data. While this largely remains a work in progress, the agreed-upon strategy is to carefully review every axis of every standard chart and to define standard visualization techniques for missing data across all of them. This approach is especially critical for *primary*, *secondary*, and *tertiary charts*, which are the ones being produced most often.

Principia Pictura Framework

Principia Pictura advocates a layered approach for the implementation of a *chart rendering engine*, constructed around a framework made of four main layers. While the implementations of each layer may vary, the structure of the framework remains the same.

4. Zomma — A grammar of visuals
3. Gamma — A grammar of charts built on top of *Principia Data*
2. Vega — A declarative visualization grammar
1. D3 — A declarative rendering engine

Appendix

Conventional Statistical Plots

The following table outlines all statistical plots used in the e-Handbook of Statistical Methods published by NIST/SEMATECH. On every row, the left column references a statistical plot, and the right column indicates how it can be produced using *Principia Pictura*.

4-Plot	Tiling
6-Plot	Tiling
Autocorrelation Plot	Range on Level (with Rule)
Bland-Altman Plot	Scatter (with References)
Bihistogram	Bar on Bar (with Mirror)
Biplot	Scatter on Scatter
Block Plot	Block
Bootstrap Plot	Tiling
Box-Cox Linearity Plot	Tiling
Box Plot	Level on Block and Candle
C-Chart	Line (with References)
Candlestick Chart	Segment on Segment
Complex Demodulation Amplitude Plot	Line
Complex Demodulation Phase Plot	Level
Conditioning Plot	Level (with Tabulation)
Control Chart	Line (with References)
Cumulative Sum Control Chart	Line (with References)
DOE Mean Plot	Candlestick (with Offset)
DOE Scatter Plot	Circle (with Tabulation)
DOE Standard Deviation Plot	Candlestick (with Offset)
Dual-Flashlight Plot	Scatter
Exponentially Weighted Moving Average Chart	Line on Range (with References)
Forest Plot	Level on Candlestick (with References)
Frequency Plot	Bar
Functional Boxplot	Line on Range

Galbraith Plot	Circular Scatter
Histogram	Bar
Lag Plot	Level
Linear Correlation Plot	Level (with Connector)
Linear Intercept Plot	Level (with Connector)
Linear Residual Standard Deviation Plot	Level (with Connector)
Linear Slope Plot	Level (with Connector)
Mean Plot	Line
Normal Probability Plot	Level (with References)
NP-Chart	Line (with References)
P-Chart	Line (with References)
P-P Plot	Line (with References)
Pareto Chart	Line on Bar
Partial Autocorrelation Plot	Level (with Rule and References)
Partial Regression Plot	Scatter (with References)
Partial Residual Plot	Scatter (with References)
Poincaré Plot	Scatter (with References)
Probability Plot	Level (with References)
Probability Plot Correlation Coefficient Plot	Level (with References)
Q-Q Plot	Level (with References)
Rank Abundance Curve	Level (with Connector)
Regression Control Chart	Line on Range (with References)
Run Sequence Plot	Level (with Rule and References)
Seasonal Subseries Plot	Line (with Tabulation and Vertical Offset)
Spectral Plot	Line
Standard Deviation Plot	Level (with References)
U-Chart	Line (with References)
Violin Plot	Area on Area (with Tabulation).
Volcano Plot	Scatter (with References)
Xbar and R Chart	Line (with References)
Xbar and S Chart	Line (with References)
Youden Plot	Level