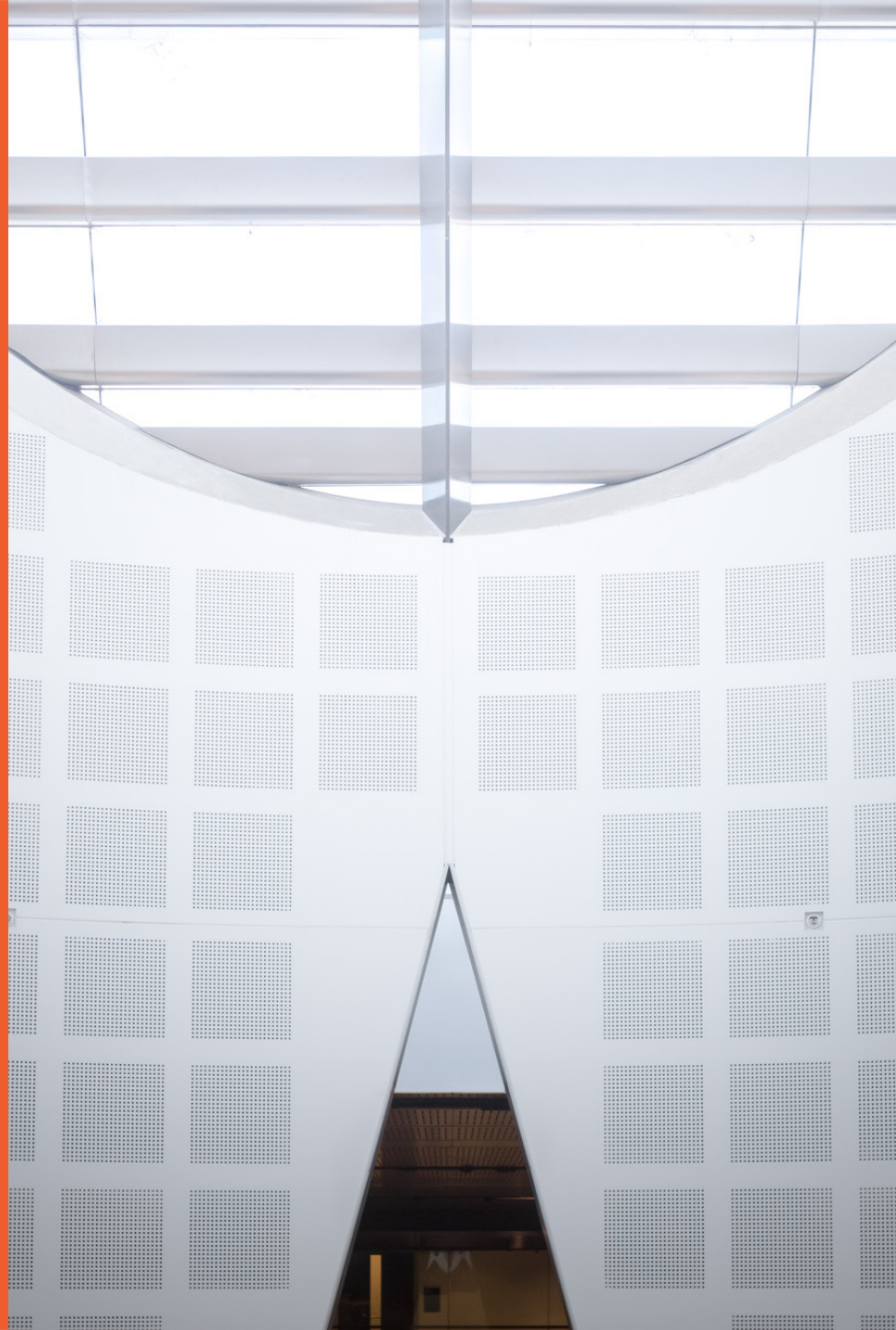
Visual Analytics in Healthcare

Dr Chang Xu

School of Computer Science

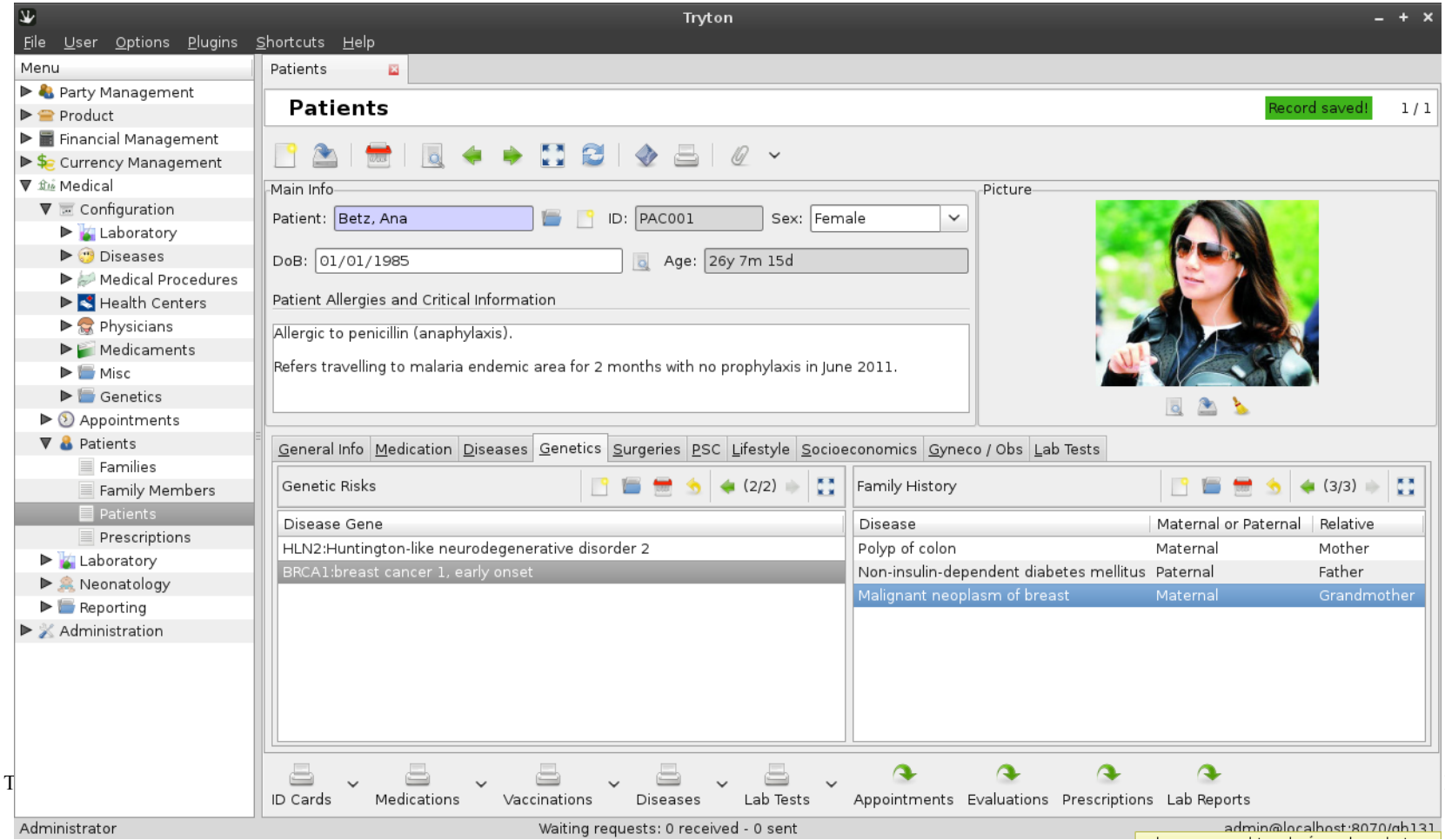
Reference: Healthcare Data Analytics, Chapter 12



THE UNIVERSITY OF SYDNEY

Clinical Data Types

The predominant data type found within **electronic health records (EHRs)** is **unstructured free text** that generally requires natural language processing (NLP) or text analysis to be standardized into a computable form.



Clinical Data Types

However, there is also **structured data** found within EHRs.

Blood Test	Result	Normal Value
WBCs (billion/L)	8.00	3.5 to 10.5
Neutrophils (%)	62	40 to 70
Lymphocytes (%)	28	25 to 45
Monocytes (%)	10	2 to 8
Eosinophils (%)	1	1 to 5
Basophils (%)	0	0 to 1
RBCs (trillion/L)	3.84	4.3 to 5.7
Hb (g/dL)	11.7	13 to 17
Hematocrit (%)	37	37 to 52
Platelets (billion/L)	262	150 to 450

Variable		Freq.	%	Variable		Freq.	%
Household Type (hhtype)				**Smoking, Cigar or Tobacco Habits (smoking)**			
1	Elementary household with one child	234	22.92	1	Yes	586	57.39
2	Elementary household with two children	245	24.00	2	No	435	42.61
3	Elementary household with three and more children	107	10.48	**Drinking Alcohol Habits (alcohol)**			
4	Childless couple	152	14.89	1	Yes	442	43.29
5	Patriarchal or extended household	135	13.22	2	No	579	56.71
6	Household with one adult	135	13.22	**Being Active (Paid) in Sports, Entertainment, Culture Etc (sport)**			
7	Students, workers etc. Living together	13	1.27	1	Yes	353	34.57
				2	No	668	65.43
Property Status (property)				**Having Private Health Insurance (insurance)**			
1	Landlord	533	52.20	1	Yes	389	38.1
2	Tenant	300	29.38	2	No	632	61.9
3	Public housing	26	2.55				
4	Not a landlord but does not pay rent	162	15.87				

Feature type	Feature name	Data type	Normal Range	UoM	Min-Mean-Max
Demographics	BMI	N	18.5 - 25	kg/m^2	20-33.117-45
Diabetes Lab Tests	HbA1C	N	<=5	mmol/L	5-6.373-7.4
Hematological Profile	Hemoglobin	N	12 - 16	g/dL	9.8-12.332-13.4
	White cell count	N	4 - 11	10^3/cmm	6-8.055-9.2
Symptoms	Urination frequency	O	-	-	-
Kidney Function Lab tests	Serum Uric acid	N	3.0 - 7.0	mg/dL	3-4.237-7.9
	Serum Creatinine	N	0.7 - 1.4	mg/dL	0.9-1.35-3.6
Lipid Profile	LDL cholesterol	N	0 - 130	mg/dL	50-94.917-170
Tumor Markers	FERRITIN	N	28 - 397	ng/mL	-
Liver Fun. Tests	S. Albumin	N	3.5 – 5.0	g/dL	1.9-4.082-5.4
Females History	Amenorrhea	O	-	-	-
Diagnosis	Diabetes Diagnosis	C	-	-	-

Clinical Data Types

However, there is also **structured data** found within EHRs that can take one of several forms:

1. **Quantitative data** refers to elements and/or measurements stored using numerical representations. These are values on which arithmetic operations can be performed such as blood test results or imaging data.
2. **Interval data** refers to data types such as date ranges (e.g., months or years) or test result intervals (above normal, normal, below normal) that include ordered ranges of quantitative measures.
3. **Ordinal data** refers to ordered measures such as classifying a patient's condition as mild, moderate, and severe
4. **Categorical data** are discretely defined nominal measures that have no inherent ordering. For example, patient gender and country of citizenship are categorical values.
5. **Hierarchical data** is that which can be represented using a tree-like structure in which the parent-child structure of the tree captures containment relationships within the data.

Introduction

In general, visual analytics tools seek to present information in ways that let the human brain detect meaningful patterns using expert knowledge.

In the clinical domain, the value and usability of the tools lies in customizing them to act as an intermediary between the multiple clinical variables, databases, and the specific goals of the clinicians or researcher.

Introduction

During the last ten years, many graphing and plotting software applications such as **Excel, SAS, SPSS, Tableau, Spotfire, and QlikView** have successfully been used to analyze complex datasets

However, most of these applications are limited when trying to illustrate the relationship between more than three variables and are not optimized for clinical and healthcare applications.

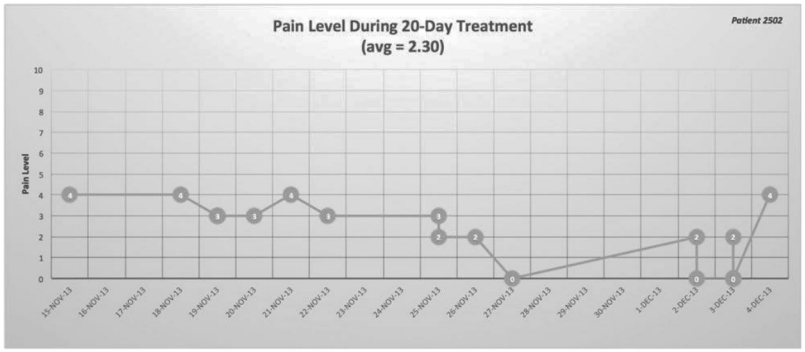
We first introduce the visualization techniques that are frequently used in clinical practice.

Standard Techniques to Visualize Medical Data

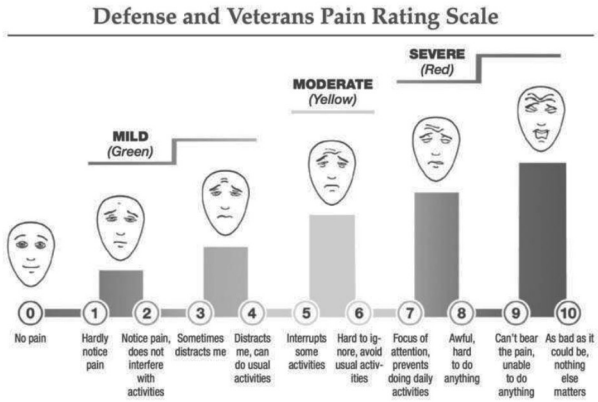
Standard Techniques to Visualize Medical Data

Date	Pain
15-Nov-13	4
18-Nov-13	4
19-Nov-13	3
19-Nov-13	3
20-Nov-13	3
21-Nov-13	4
22-Nov-13	3
25-Nov-13	3
25-Nov-13	2
26-Nov-13	2
27-Nov-13	0
2-Dec-13	2
2-Dec-13	0
3-Dec-13	0
3-Dec-13	2
3-Dec-13	0
4-Dec-13	4

(a)



(b)



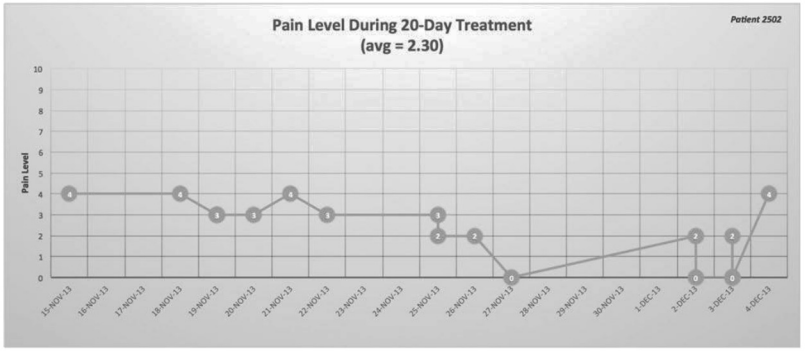
(c)

Electronic medical records (EMRs), clinical devices, and many software applications used in hospital organizations provide basic plotting capabilities including histograms, line plots, pie charts, scatter plots, and other techniques to display clinical data.

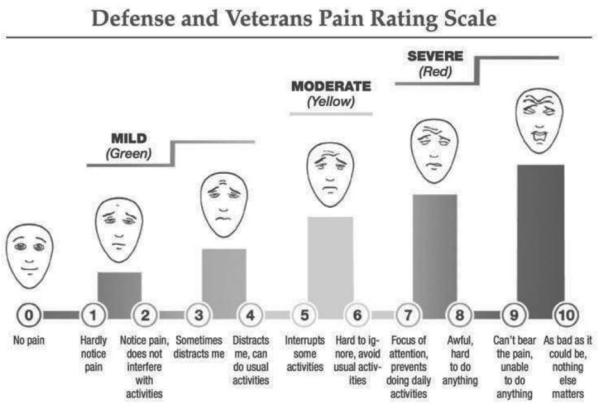
Standard Techniques to Visualize Medical Data

Date	Pain
15-Nov-13	4
18-Nov-13	4
19-Nov-13	3
19-Nov-13	3
20-Nov-13	3
21-Nov-13	4
22-Nov-13	3
25-Nov-13	3
25-Nov-13	2
26-Nov-13	2
27-Nov-13	0
2-Dec-13	2
2-Dec-13	0
3-Dec-13	0
3-Dec-13	2
3-Dec-13	0
4-Dec-13	4

(a)



(b)



(c)

To overcome some of those limitations, tables often support features such as sorting, filtering, and coloring.

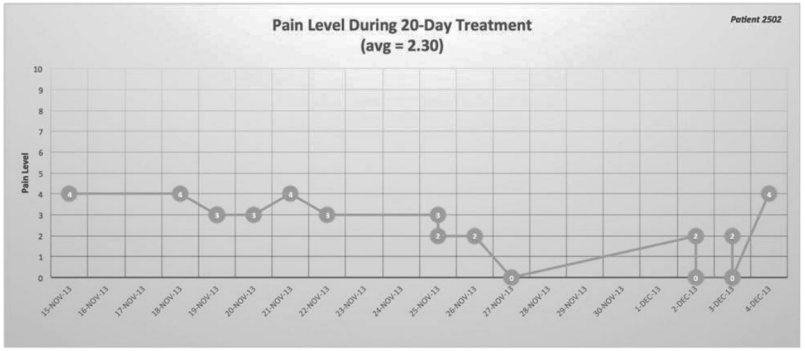
Figure a shows the pain scale values for a given patient that went through an intensive 20-day treatment. A colormap is a way to show the data to quickly illustrate a pattern.

The most common technique to show structured clinical data within EHRs are tables. Unfortunately, one of the key limitations of using tables to illustrate clinical data is that as the number of columns and rows increases, the reader quickly becomes **overwhelmed with the amount of information.**

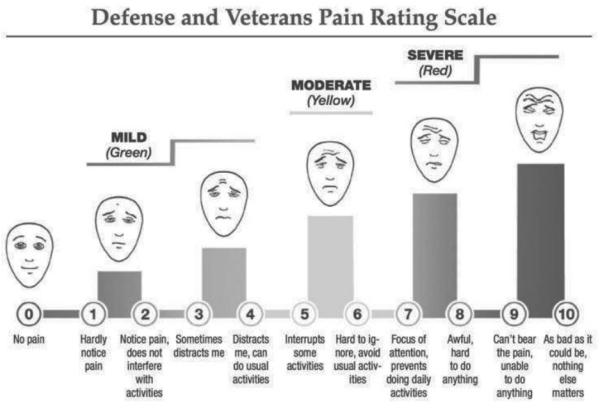
Standard Techniques to Visualize Medical Data

Date	Pain
15-Nov-13	4
18-Nov-13	4
19-Nov-13	3
19-Nov-13	3
20-Nov-13	3
21-Nov-13	4
22-Nov-13	3
25-Nov-13	3
25-Nov-13	2
26-Nov-13	2
27-Nov-13	0
2-Dec-13	2
2-Dec-13	0
3-Dec-13	0
3-Dec-13	2
3-Dec-13	0
4-Dec-13	4

(a)



(b)



(c)

Figure b shows the pain scale for a given patient that went through an intensive 20-day treatment. This plot allows referring providers to better understand how the treatment is affecting the patient's overall pain.

Figure c is the illustration of the DoD/VA pain rating scale shown to patients to better standardize pain assessments.

Among plotting techniques, line charts are the most common visualization technique available within electronic health records (EHRs). Even though there are differences between EHR vendors, the vital signs tab for most platforms allows the plotting of measurements such a body temperature, heart rate, and blood pressure, among other elements.

Standard Techniques to Visualize Medical Data

When basic plotting techniques like line plots incorporate dynamic and interactive interfaces, users become more engaged with the data analysis and are able to obtain a significant amount of insight about the data more quickly.

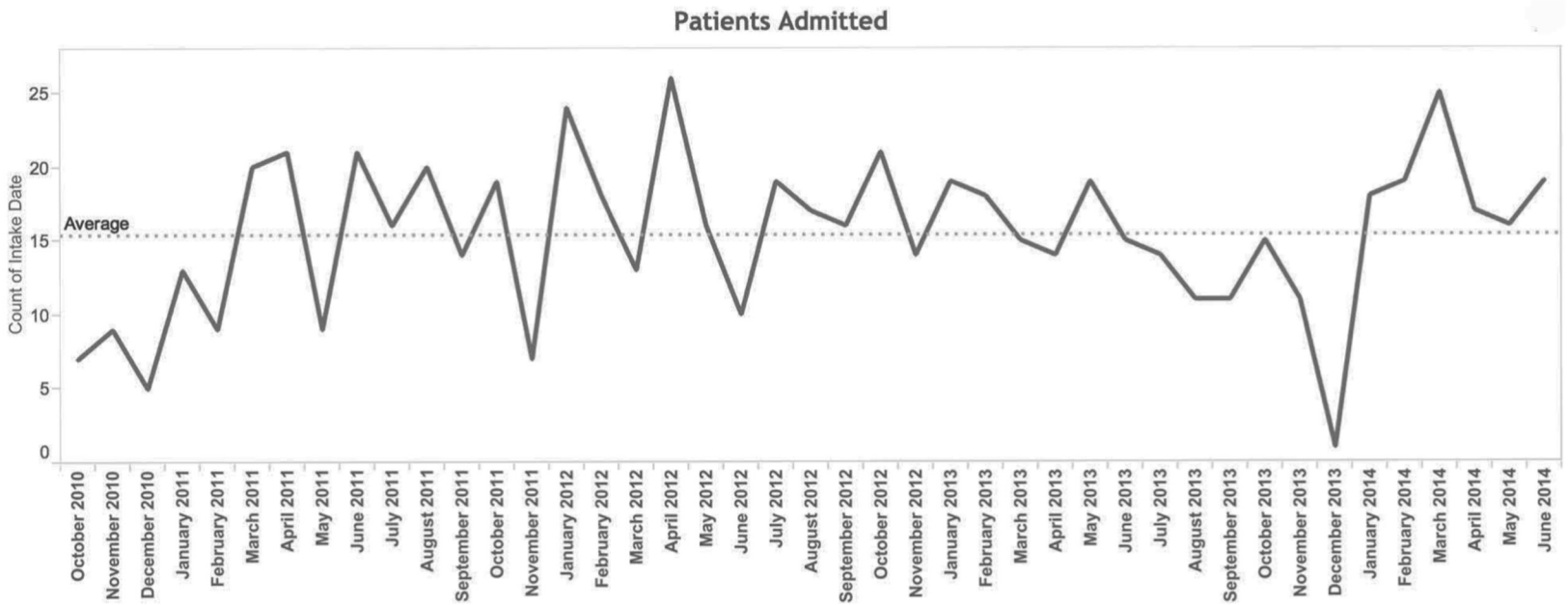


Figure shows a line plot illustrating the number of specific patients that a particular provider admitted during a 45-month period. Such a plot is used to see patterns, measure productivity, and justify additional clinical resources.

Standard Techniques to Visualize Medical Data

In general, basic visualization techniques are used to display a single variable or a small set of elements that are of the same type and range. However, multiple basic charting techniques can be incorporated into a single plot, thus allowing the illustration of multiple variables or data types simultaneously.

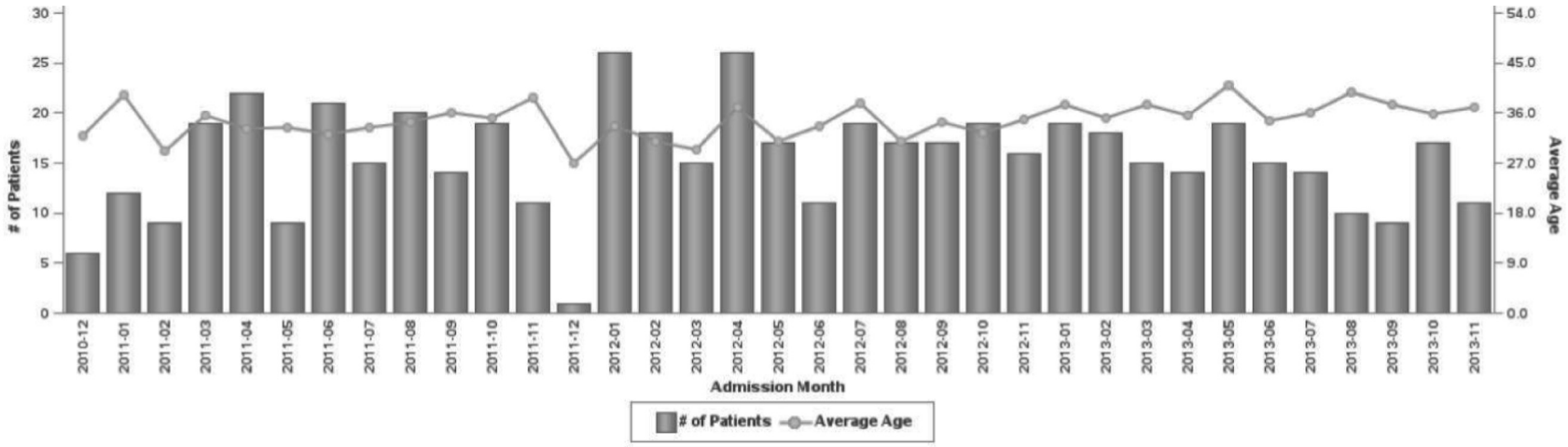


Figure combines bar charts and line plots. The bar charts are used to illustrate the number of specific patients that visited a provider during a 36-month period and a line chart in the right axis illustrates the average age of those patients.

Standard Techniques to Visualize Medical Data

Another commonly used approach to display multiple variables within a single chart is by doing a stacked plot.

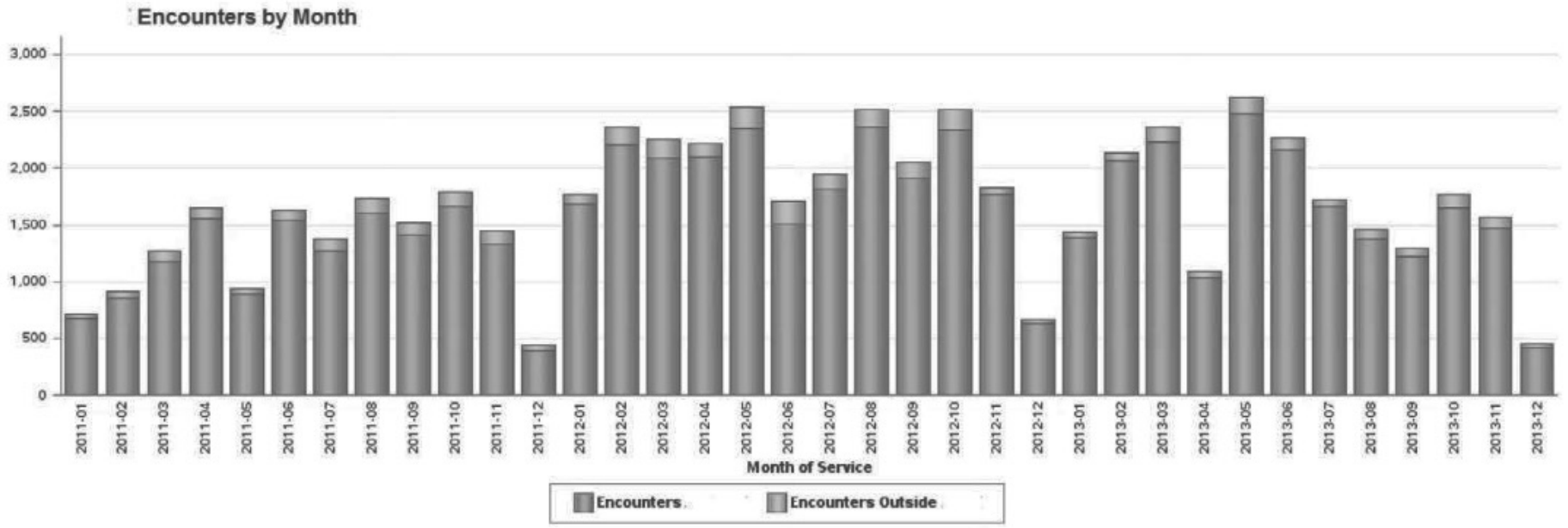


Figure shows a stacked plot illustrating the monthly patient encounters that took place within a given department versus those encounters where patients were transferred to a different location. By stacking the variables within a single chart the user can obtain three measurements: (a) the number of encounter per month within the hospital department under consideration, (b) the number of encounters per month that require a patient's transfer, and (c) **the total number** of encounters providers of the department under consideration are involved in. In addition, the temporal aspect of the chart helps users to analyze the changes of each of those three measurements over time.

High-Dimensional Data Visualization

High-Dimensional Data Visualization

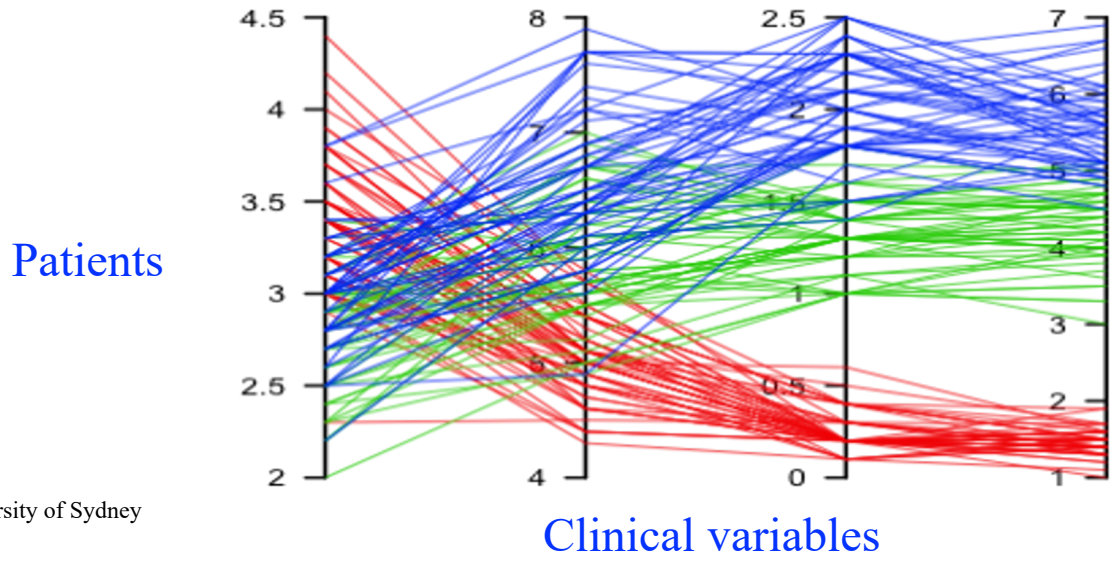
Over the last several years, multiple high-dimensional data visualization techniques have been proposed including **parallel coordinates**, **star glyphs**, **tree graphs**, **treemaps**, and **dependency graphs**, to enable interactive exploration of complex clinical datasets and help users identify previously unknown patterns

Parallel coordinate

Parallel coordinates are a powerful method for visualizing multidimensional data.

Given an N × M spreadsheet with N patients containing M clinical variables, a parallel coordinate visualization is created by displaying M equally spaced vertical axes with individual ranges.

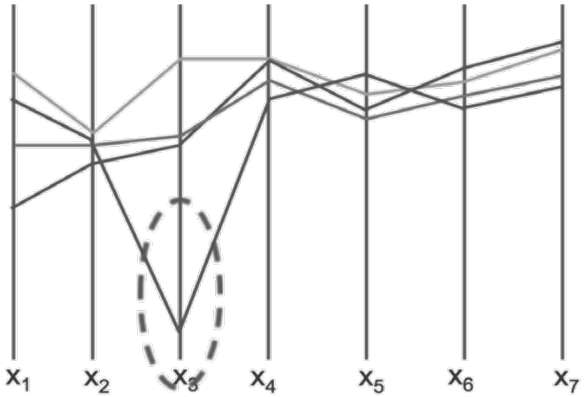
Each of the N patients are illustrated as a line that passes through each of the axes. Once generated, the visualization technique allows the users to interactively define a range within a single or multiple variables (axes) and explore correlations between variables for the selected patients.



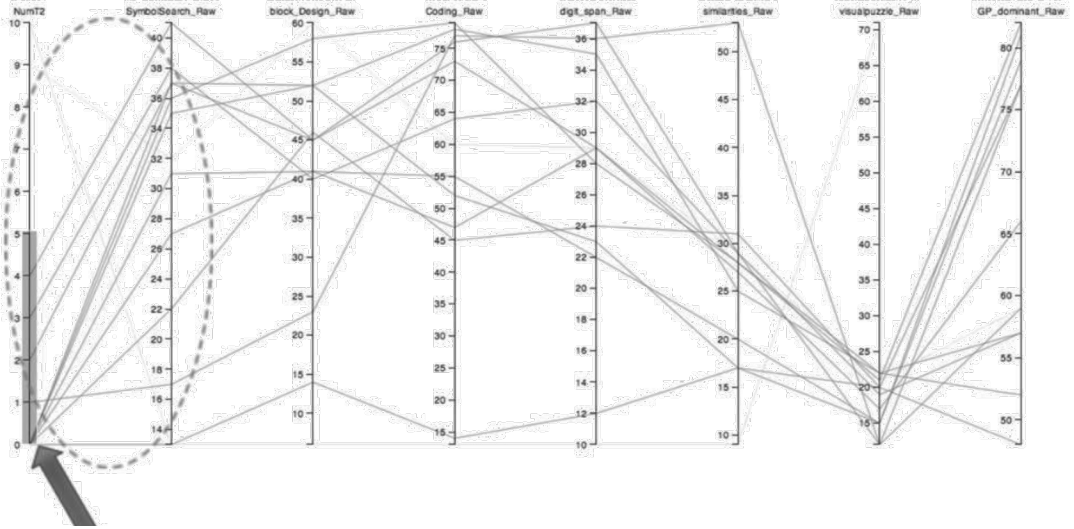
Parallel coordinate

Figure (a) shows a diagram illustrating how parallel coordinates plots are created and used. This particular example is to show how the human eye can quickly look at multiple variables and immediately find sample lines (e.g., patients) that are outside the normal range.

Figure (b) shows results of using parallel coordinates to explore the relationship between imaging and neuropsychological measurements.



(a)

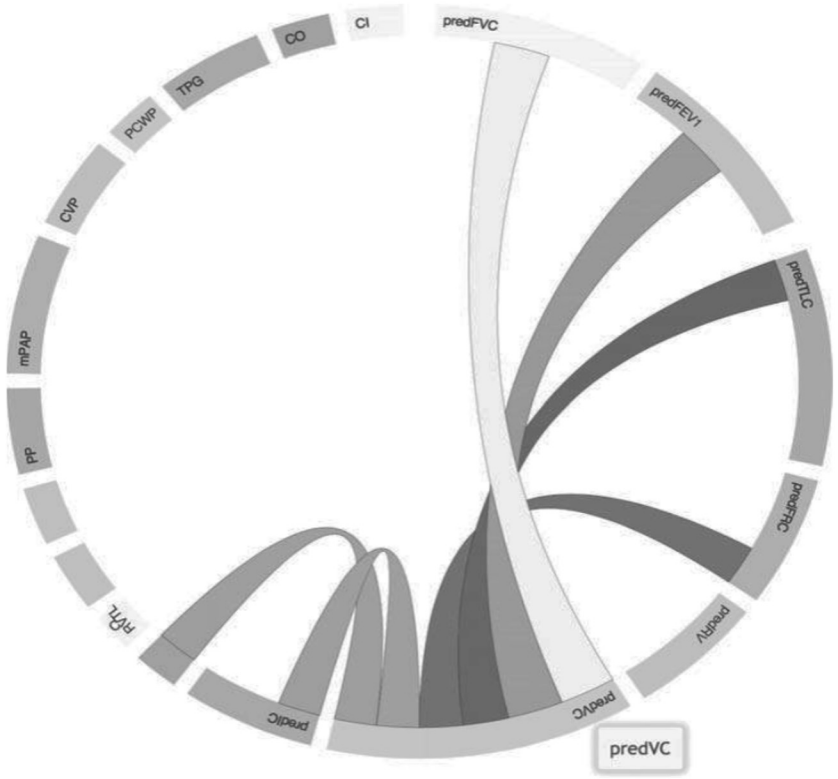


If we compare the first axis describing the number of brain lesions with the second axis describing the neuropsychology test "Symbol Search," we can see that most of the relationships between these two variables are linear.

Chord visualization plot

Another technique to analyze the relationship between many variables is by using a chord visualization plot.

These plots are generated by creating a circular plot consisting of boundaries, where each boundary is a variable. Connections are drawn between boundaries and the thickness of each connection represents the strength of the correlation.

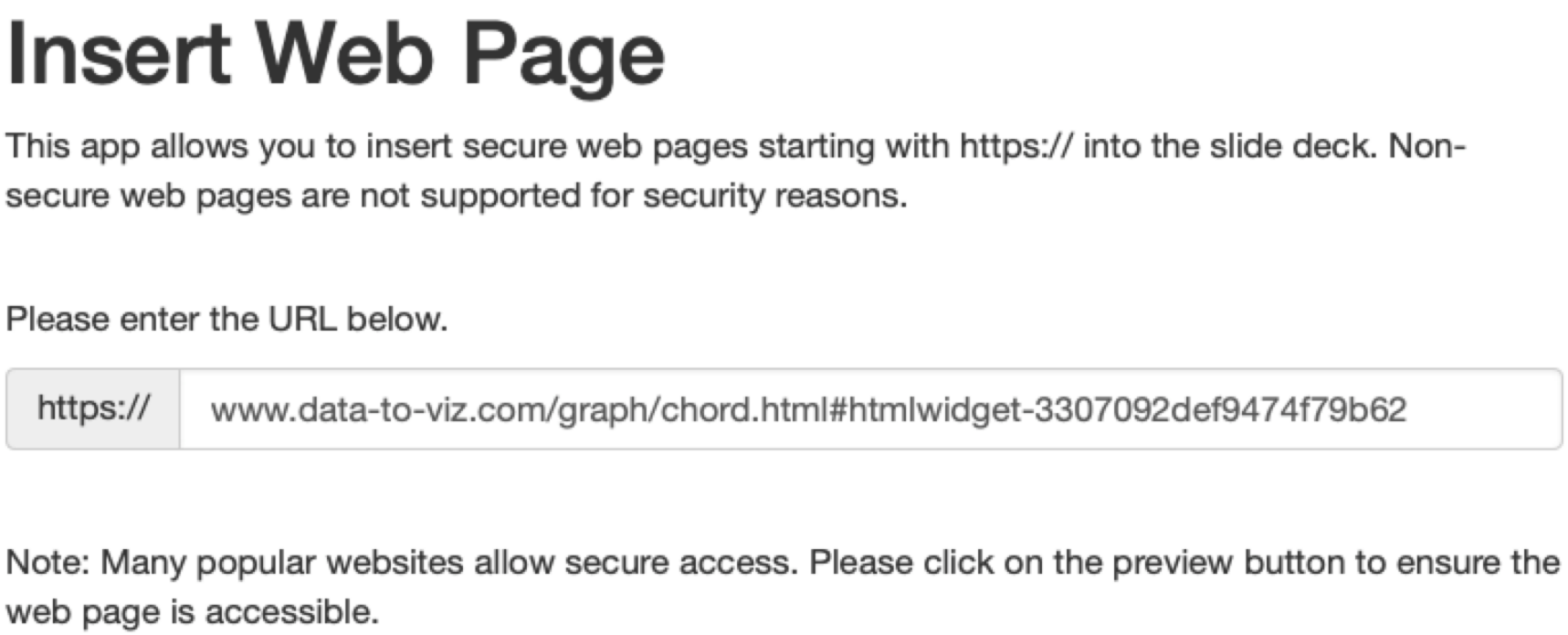


(a)

FIGURE (a) Chord visualization technique can be used to study the associations between multiple clinical variables. In this particular example, 18 clinical elements are displayed and the associations between predictive volume capacity (predVC) of the lungs and other clinical elements is being illustrated.

Chord visualization plot

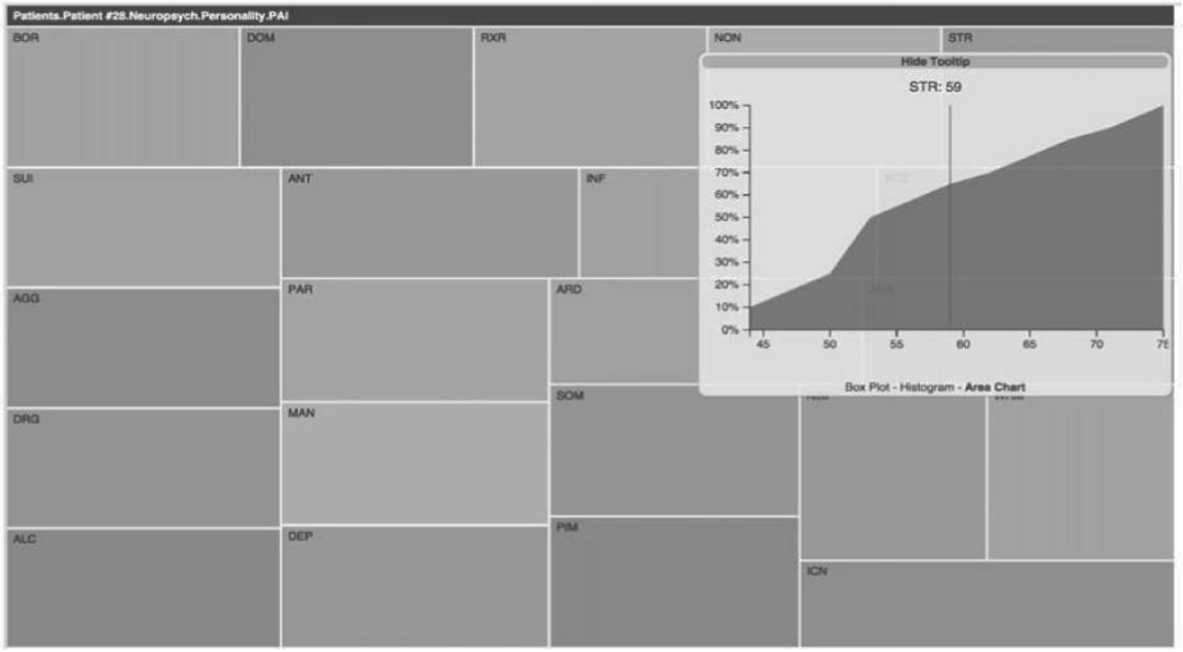
Interactivity is a real plus to make the chord diagram understandable. In the example below, you can hover a specific group to highlight all its connections.
www.data-to-viz.com/graph/chord.html#htmlwidget-3307092def9474f79b62



Hierarchical trees and treemaps.

Two of the most widely used techniques to explore hierarchical data are hierarchical trees and treemaps.

The treemap visualization places the top layer of the data (e.g., patients) in a grid as individual rectangles. The area of the rectangle corresponds to a given measurement, such as the height of the subtree or the amount of data under that specific node.



Personality Assessment Inventory (PAI) in neuropsychological testing: Somatic Complaints (SOM), Anxiety (ANX), Depression (DEP), Borderline Features (BOR), etc.

Hierarchical trees and treemaps.

In the figure below, clicking on a group zooms on it a reveals the underlying structure. Hint: click on the title to come back to the previous level of the hierarchy.
www.data-to-viz.com/graph/treemap.html#htmlwidget-40fb4aa85a5a5e4a2487

Curse of dimensionality

Most tools visualize high-dimensional data by giving each data point a location in a 2 or 3-dimensional map and leave the interpretation of the data to the human observer.

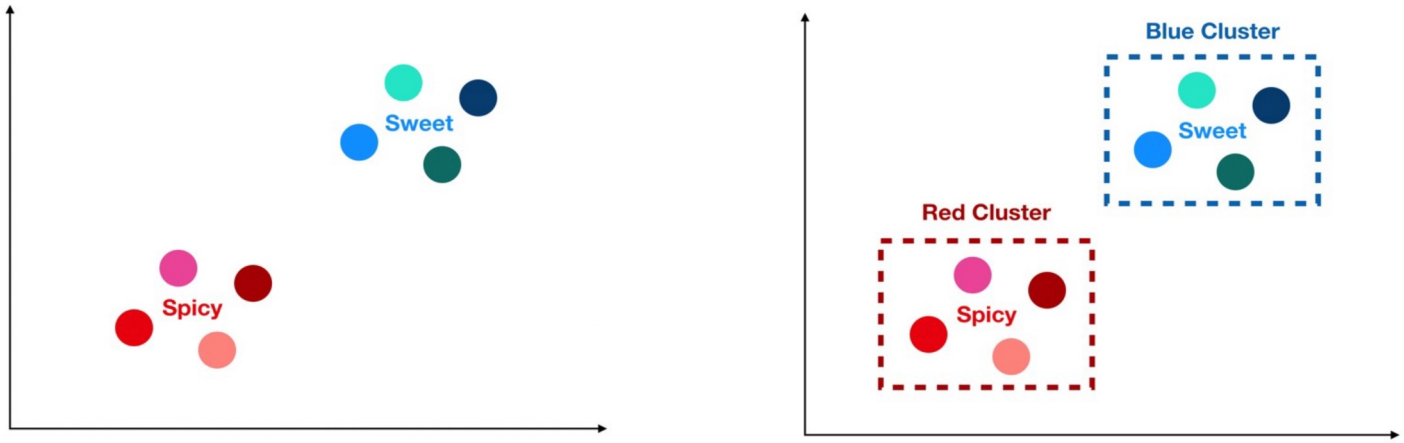
Working with large data presents many challenges, one of them being a loss of efficiency and performance in your models due to too high dimensionality.

Curse of dimensionality - Toy Example

Imagine our dataset consists of the following 8 candies.

The ground truth is that there are two clusters within our dataset of 8 candies — spicy and sweet.

Thanks to **clustering**, if we eat a reddish candy, it will be spicy; and if we eat a bluish candy, it will be sweet.



Curse of dimensionality - Toy Example

What if our data is high dimensional like the table on the right?

Every candy is its own color and we have gained **zero insight** into how to predict whether a given candy will be spicy or sweet.

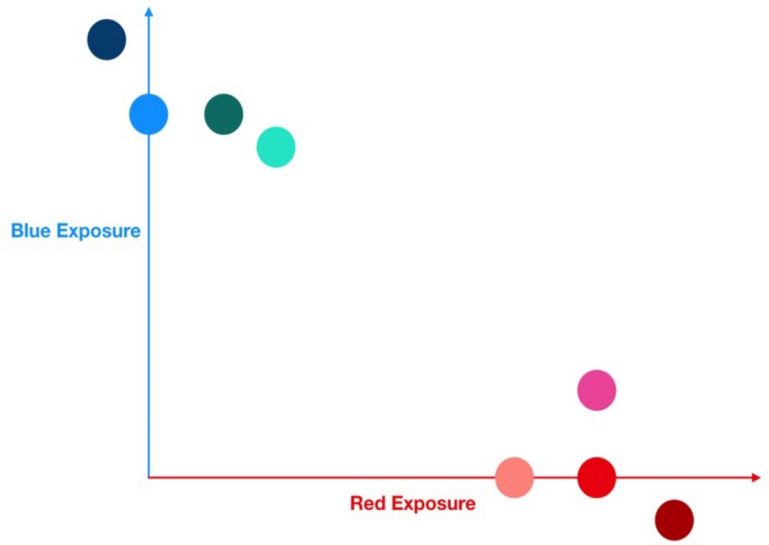
	Red	Maroon	Pink	Flamingo	Blue	Turquoise	Seaweed	Ocean
	1	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0
	0	0	1	0	0	0	0	0
	0	0	0	1	0	0	0	0
	0	0	0	0	1	0	0	0
	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	1

Dimensionality Reduction to the Rescue

The dimensionality reduction algorithm starts by locating the underlying trends in our features

For our candy dataset, the underlying trends would most likely be the primary colors red and blue.

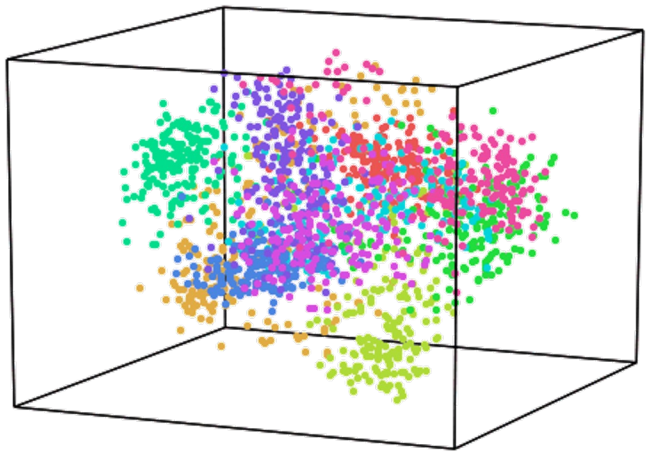
	Red	Blue
Red	1.00	0
Maroon	1.20	-0.10
Pink	1.00	0.20
Flamingo	0.80	0
Blue	0	1.00
Turquoise	0.25	0.90
Seaweed	0.15	1.00
Ocean	-0.10	1.20



Dimensionality Reduction

Luckily, many dimensionality reduction techniques are available that can help us overcome challenges by enabling us to remove “less important” data.

The aim of dimensionality reduction is to preserve as much of the significant structure of the high-dimensional data as possible in the low-dimensional map.



Dimensionality Reduction

Dimensionality reduction sits under the Unsupervised branch of Machine Learning algorithms.

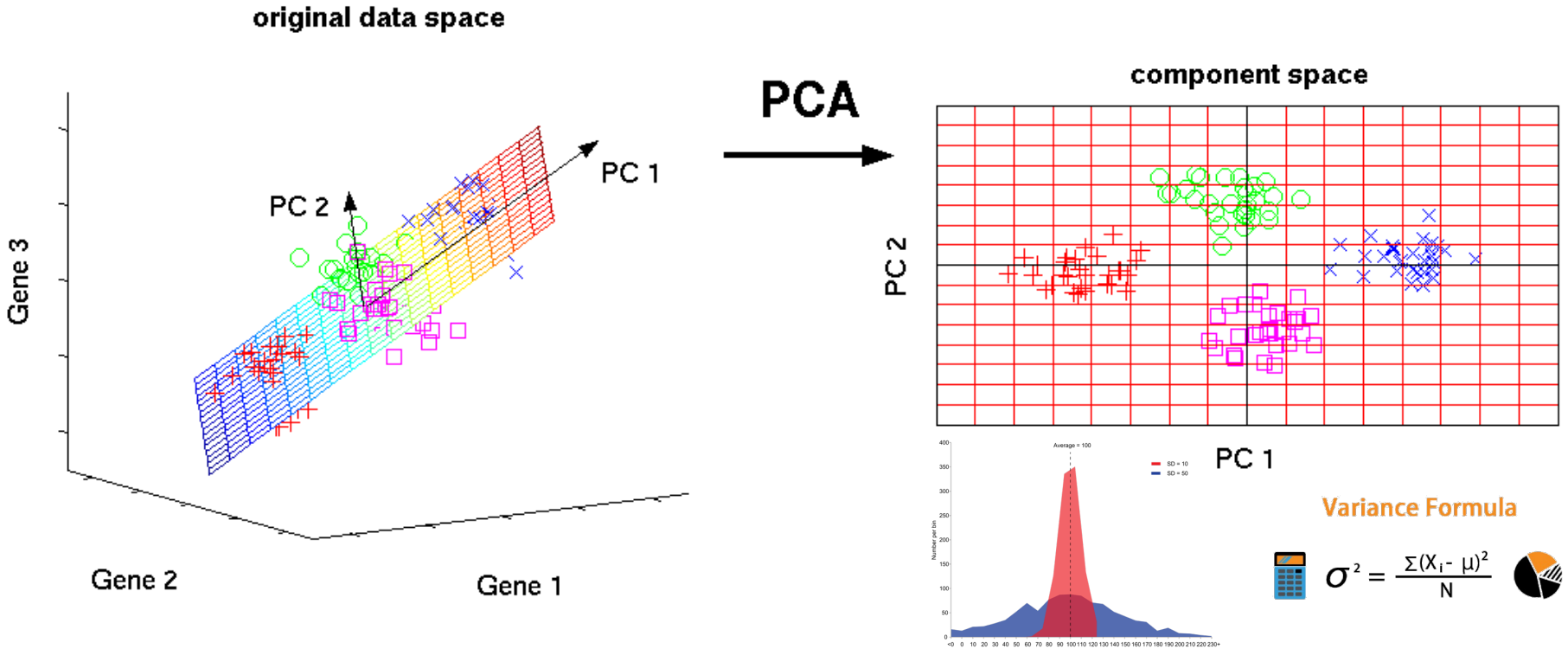
Two popular dimensionality reduction algorithms:

- Principal Component analysis (PCA)
- t-distributed stochastic neighborhood embedding (t-SNE)

Principal Component analysis (PCA)

PCA is an unsupervised **linear dimensionality reduction** and data visualization technique for very high dimensional data.

Project the data onto a lower dimensional space such that the variance of the projected data is maximized.

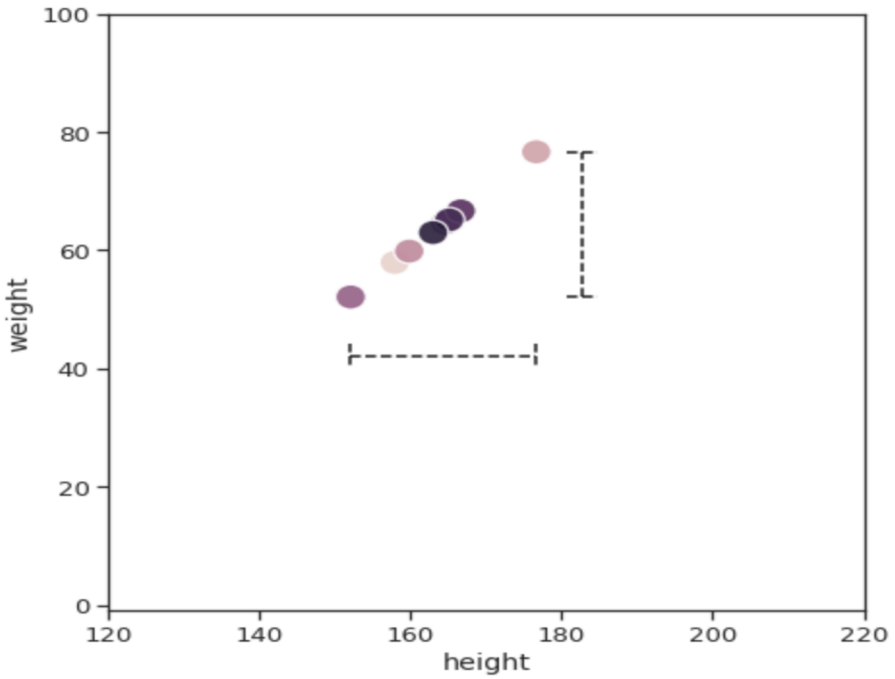


Principal Component analysis (PCA)

In the eyes of PCA, **Variance is information**.

How PCA summarizes our data, or more accurately, reduces dimensionality?

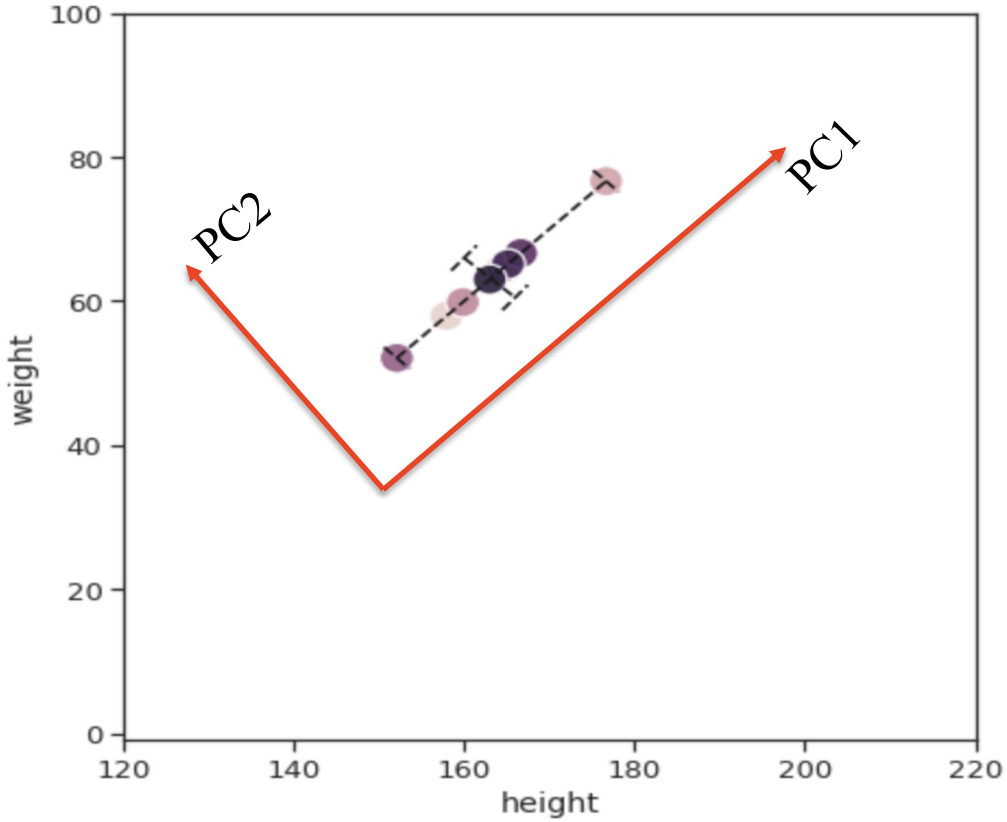
Example: height and weight data for 7 people



Feature	Variance
Height	1.11
Weight	1.11
Total	2.22

Principal Component analysis (PCA)

the maximum amount of variance lies not in the x-axis, not in the y-axis, but **a diagonal line across**. The second-largest variance would be a line 90 degrees that cuts through the first.



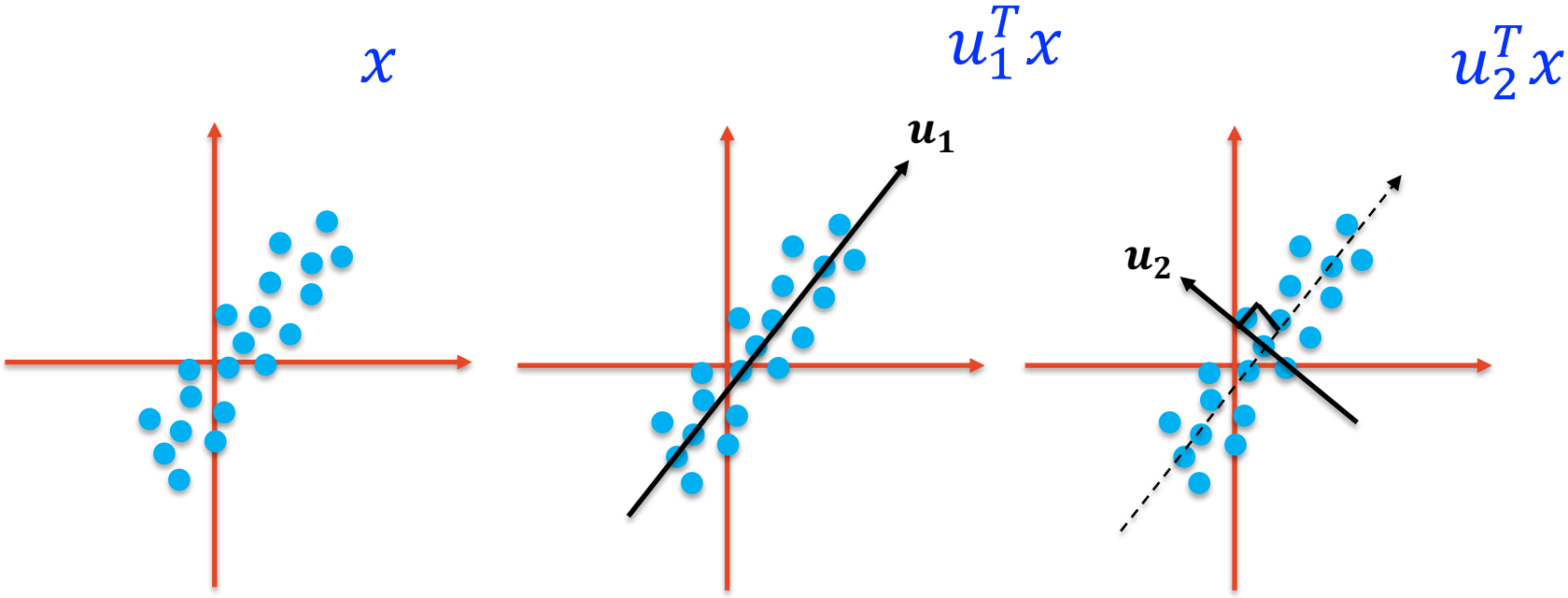
Feature	Variance
PC1	2.22
PC2	0.00
Total	2.22

PC1 alone can capture the total variance of Height and Weight combined.

Since PC1 has all the information, we can remove PC2 and know that our new data is still representative of the original data.

Principal Component Analysis (PCA)

Maximum Variance Formulation



data

The first principal component

The second principal component

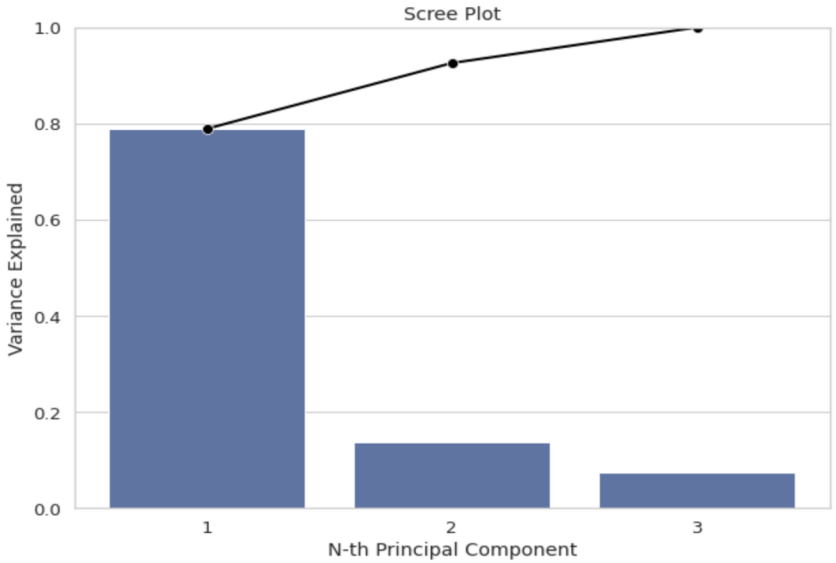
Principal Component analysis (PCA)

When it comes to real data, more often than not, we won't get a principal component that captures 100% of the variances.

Performing a PCA will give us N number of principal components, where N is equal to the dimensionality of our original data.

From this list of principal components, we generally choose the least number of principal components that would explain the most amount of our original data.

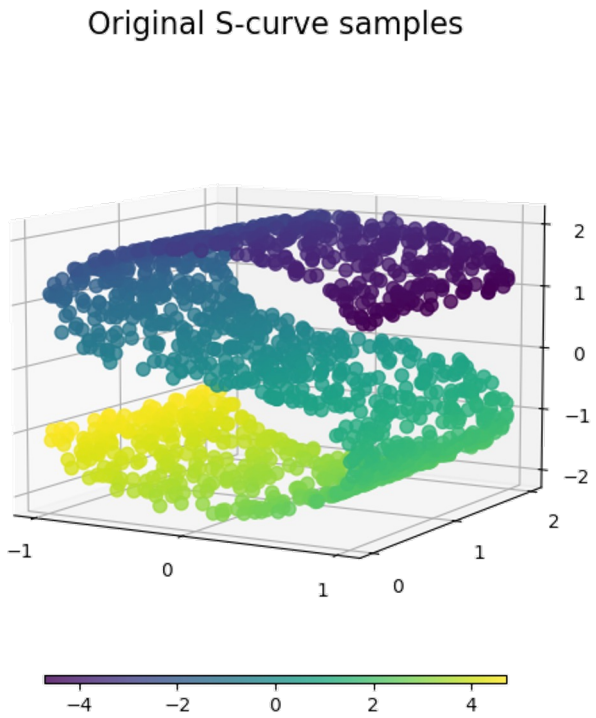
A great visual aid that will help us make this decision is a Scree Plot.

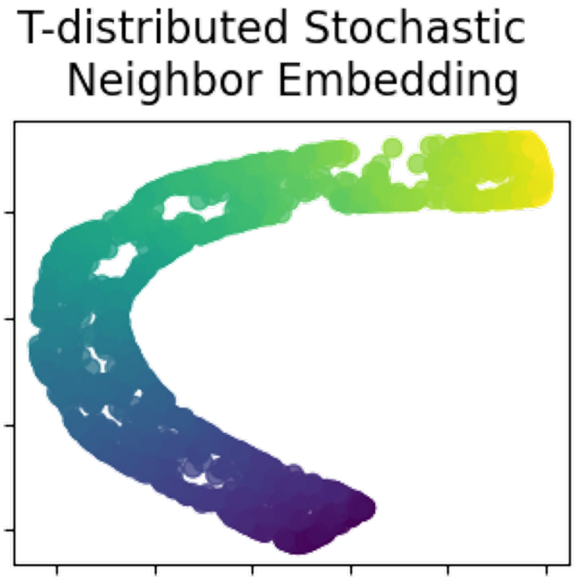


t-distributed stochastic neighbourhood embedding (t-SNE)

PCA is a linear dimension reduction technique.

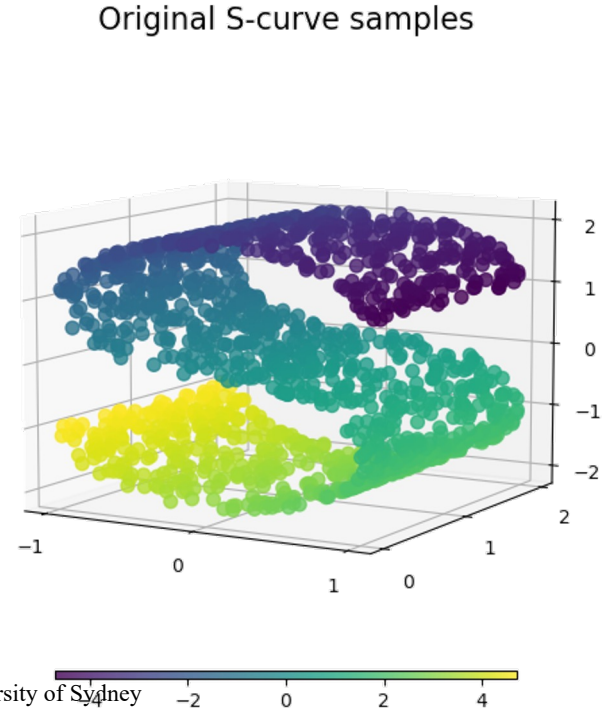
This can lead to poor visualization especially when dealing with non-linear manifold structures as any geometric shape like: cylinder, ball, curve, etc.

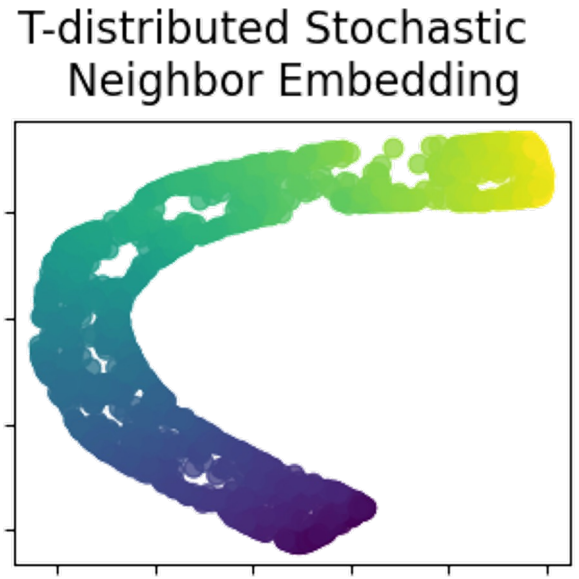




t-distributed stochastic neighbourhood embedding (t-SNE)

The t-SNE algorithm calculates a similarity measure between pairs of instances in the high dimensional space and in the low dimensional space. It then tries to optimize these two similarity measures using a cost function.



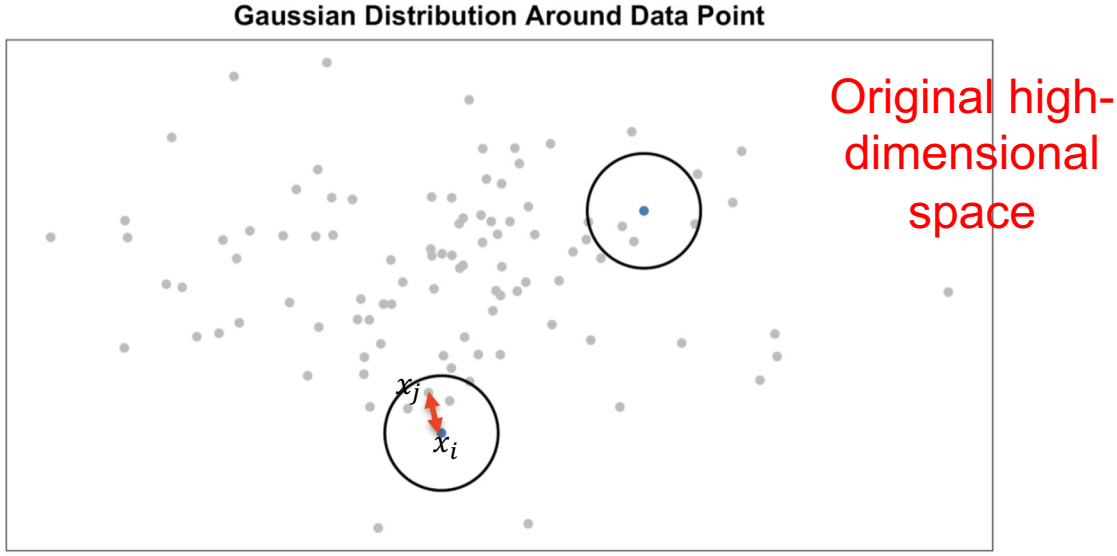


t-distributed stochastic neighbourhood embedding (t-SNE)

t-SNE: Step 1

Think of a bunch of data points scattered on a 2D space. For each data point (x_i) we'll center a Gaussian distribution over that point.

Then we measure the density of all points (x_j) under that Gaussian distribution, which gives us a set of probabilities (P_{ij}) for all points. Those probabilities are proportional to the similarities.



t-distributed stochastic neighbourhood embedding (t-SNE)

t-SNE: Step 2

Step 2 is similar to step 1, but instead of using a Gaussian distribution you use a Student t-distribution, which is also known as the Cauchy distribution.

This gives us a second set of probabilities (Q_{ij}) in the low dimensional space. As you can see the Student t-distribution has heavier tails than the normal distribution. The heavy tails allow for better modeling of far apart distances.

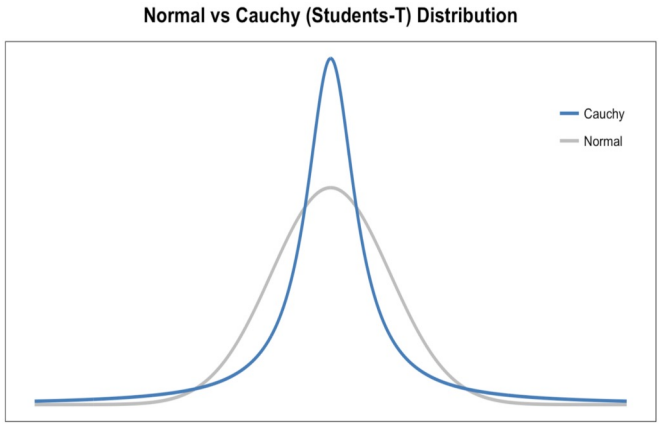
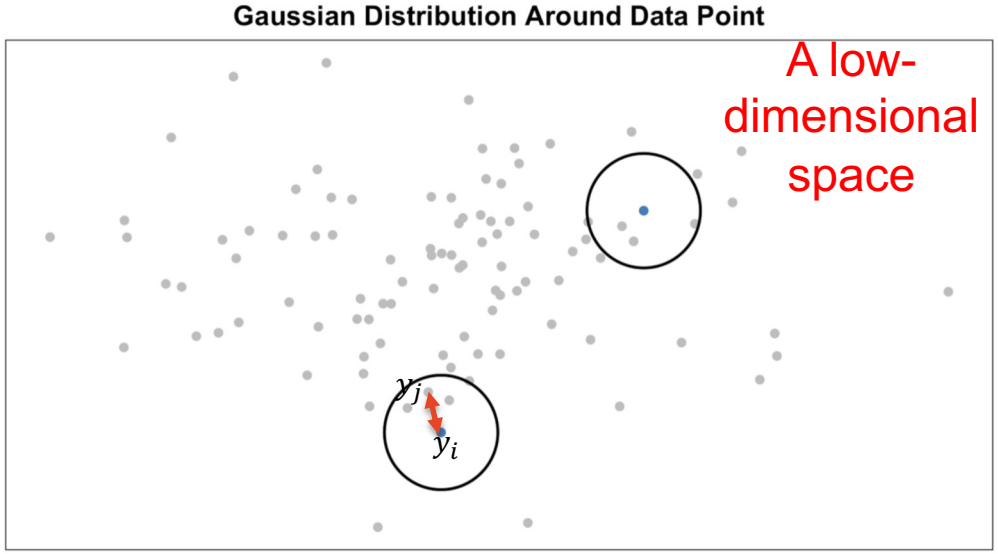


Figure 3 — Normal vs Student t-distribution



t-distributed stochastic neighbourhood embedding (t-SNE)

t-SNE: Step 3

The last step is that we want these set of probabilities from the low-dimensional space (Q_{ij}) to reflect those of the high dimensional space (P_{ij}) as best as possible.

We want the two map structures to be similar. We measure the difference between the probability distributions of the two-dimensional spaces using Kullback-Liebler divergence (KL).

Finally, we use gradient descent to minimize our KL cost function.

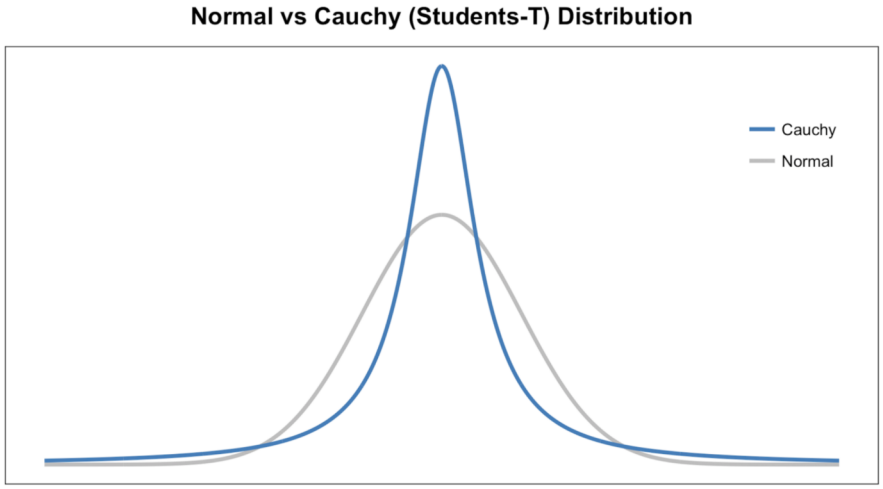


Figure 3 — Normal vs Student t-distribution

t-SNE vs. PCA

t-SNE	PCA
t-SNE is nonlinear dimensionality reduction method	PCA is linear dimensionality reduction method
t-SNE fails to preserve the global geometry but produces well-separated clusters. Keeping large perplexity parameter can help to preserve the global geometry of data	PCA preserves the global data structure but fail to preserve the similarities within the clusters
t-SNE is computationally expensive and can take several hours on large dataset	PCA is much faster than t-SNE for large datasets. It is recommended to run PCA before running t-SNE to reduce the number of original variables.
t-SNE is a stochastic method and produces slightly different embeddings if run multiple times	It is not necessary to run PCA multiple times
Several parameters (hyperparameters) such as perplexity and learning rate needs to optimized based on the datasets	Generally no parameter needs to optimized in PCA

* The higher the perplexity is the higher value variance has.