Make any stupid plot you want

Hannah Aizenman

A dissertation proposal submitted to
the Graduate Faculty in Computer Science in partial fulfillment of the
requirements for the degree of Doctor of Philosophy,
The City University of New York

Committee Members:
Dr. Michael Grossberg (Advisor), Dr. Robert Haralick, Dr. Lev Manovich,
Dr. Huy Vo

Abstract

Contents

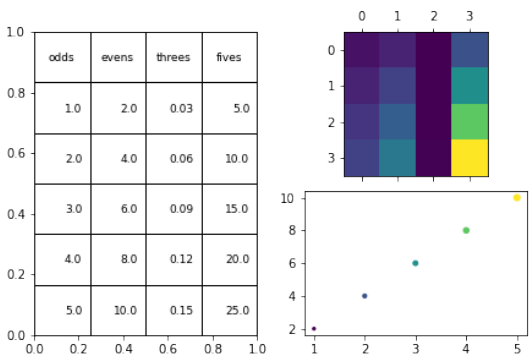


Figure 1: Implicit in visualization is the assumption that these three representations of data are equivalent, specifically that the measurements within a variable and relations of the measurements of each variable are preserved.

1 Introduction

1.1 Thesis statement

We define a visualization as a transform from data to graphic that preserves the topology of the data and faithfully map the properties of the measurement type. In fig 1, we implicitly assume that the translation from table to heatmap has preserved the order of observations (the rows) and that the perceptually uniform sequential colormap has been applied such that the ordering relation on floats matches the ordering on the colormap (darker colors map to larger numbers). We also make this assumption about color in the scatter map, and that the translation to size and position on screen also respect the ordering on floats. In this work, we propose to mathematically describe the transform of data to visual space such that we can make explicit the implicit topology and types visualizations preserve. We then propose a new architecture for the Python visualization library Matplotlib [8] based on these descriptions because the Matplolib artist layer is analogous to the transforms.

1.2 What is a viz

? Acquired codes of meaning

2 Not all data are tables

Tables, images (Lev), graphs (network X)

set up: dubois

theorists: bertin, munzner, mackinlay

talk about: matplotlib arch paper, excel/matlab arch, vtk & ggplot (compare/contrast, we're blending these things)

3 Notation & Definitions

In this section we introduce a mathematical description of the visualization pipeline where artist A functions transform data of type $\Gamma(E)$ to an intermediate representation in prerendered display space of type $\Gamma(H)$:

$$A : \Gamma(E) \rightarrow \Gamma(H) \tag{1}$$

- A is the function that converts an instance of data $\Gamma(E)$ to an instance of a visual representation $\Gamma(H)$
- E is a locally trivial fiber bundle over K representing data space.
- K is a triangulizable space encoding the connectivity of the points in E
- H is a fiber bundle over S representing visual space
- S is a simplacial complex encoding the visualization
- Γ is the global space of sections

When E is a trivial fiber bundle $E = F \times K$, it can be assumed that all fibers F_k over $k \in K$ are equal. Fiber bundles are product spaces of toplological spaces, which are a set of points with a set of neighborhoods for each point[7, 16].

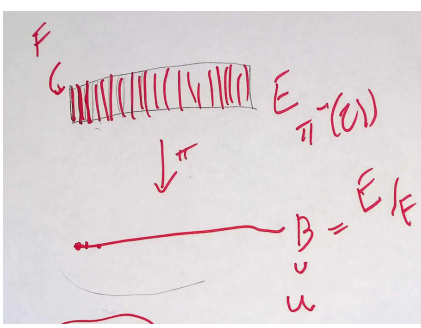
3.1 Fiber Bundles

We provide a brief description of fiber bundles because we model data, visual transformations, and a prerendered visual graphic as fiber bundles. A fiber bundle is a structure (E, B, π, F) consisting of topological total space E, base space K, fiber space F and the map from total space to base space:

$$F \hookrightarrow E \xrightarrow{\pi} B \tag{2}$$

where there is a bijection from F to every fiber F_b over point $b \in B$ in E and the function $\pi : E \to B$ is the map into the B quotient space of E. By defintion of a fiber bundle, π is always a mapping from total space to base space, independent of the points $p \in E$, and therefore we call this mapping π for all the fiber bundles in the model.

3.1.1 Base space



B is the quotient space of E, meaning it is the set of equivalence classes of elements p in E defined via the map $\pi : E \to B$ that sends each $p \in E$ to its equivalence class in $[p] \in B$ [14].

As shown in figure **??**, the fibers F divide E into smaller spaces consisisting of F and an open set neighborhood around F. This subdivision is projected down to the toplology $\mathcal{T}$

$$\mathcal{T}_b = \{U \subseteqq B : \{p \in E : [p] \in U\} \in \mathcal{T}_E\} \tag{3}$$

where $[p] \in U$ is the point $b \in B$ with an open set surrounding it that has an open preimage in E under the surjective map $\pi : p \to [p]$.

3.1.2 Fiber

As shown in equation!**??**, every point $b \in B$ has a local open set neighborhood U [7, 16]

$$\begin{tikzcd}
\pi^{-1}(U) \arrow[r, "\varphi"] \arrow[d, "\pi"'] & U \times F \arrow[dl, "\mathrm{proj}_U"] \\
U &
\end{tikzcd} \tag{4}$$

such that $\varphi : \pi^{-1}(U) \to U \times F$ is a homeomorphism where π and proj_U both map to U and the fiber over k $F_b = \pi^{-1}(b \in B)$ is homomorphic to the fiber F.

3.1.3 Section

The section f is the mapping from base space to total space $f : B \to E$

$$\begin{tikzcd}
F \arrow[r, hook] & E \arrow[d, "\pi"'] \\
& B \arrow[u, bend right, "f"']
\end{tikzcd} \tag{5}$$

such that f is the right inverse of π

$$\pi(f(b)) = b \text{ for all } b \in B \tag{6}$$

In a locally trivial fiber bundle, $E = B \times F$ [7, 16]:

$$f(b) = (b, g(b)) \tag{7}$$

where the domain of $g(b)$ is F_b and returns a point p in F_b. The space of all possible sections f of E is $\Gamma(E)$. All sections $f \in \Gamma(E)$ have the same fibers F and connectivity B.

3.1.4 Sheaf and Stalk

As described in equation **??**, there is a local space $U \subset B$ around every b. The inclusion map $\iota : U \to B$ can be pulled back such that $\iota^* E$ is the space of E restricted over U.

$$\begin{tikzcd} \iota^* E \arrow[d, "\pi"'] & E \arrow[l, "\iota^*"'] \arrow[d, "\pi"'] \\ U \arrow[u, bend right, "\iota^* f"'] & B \arrow[l, "\iota"] \arrow[u, bend right, "f"'] \end{tikzcd} \tag{8}$$

The localized section of fibers $\iota^* f : U \to \iota^* E$ is the sheaf $\mathcal{O}(E)$ with germ of $\xi^* f$. The neighborhood of points the sheaf lies over is the stalk $\mathcal{F}_b$ [18, 20]

$$\iota^{-1}\mathcal{F}(\{b\}) = \varinjlim_{b \subseteq U} \mathcal{F}(U) = \varinjlim_{b \in U} = \mathcal{F}_b \tag{9}$$

which through ι gets the data in E at and near to b. Restricting the artist to the sheaf means the artist knows the data in F and also has access to derivatives of the data. This property is useful for some visual transformations.

3.2 Data Model

As proposed by Butler [3, 4], we model data as a fiber bundle (E, K, π, F) with $\pi : E \to K$ where K which can be thought of as a set of keys k. A section $\tau : K \to E$ is an instance of the data that lies in E and is discussed in section 3.2.4.

3.2.1 Example

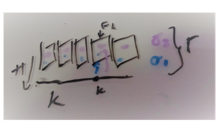


Figure 2: write up some words here

As illustrated by figure **??**, the vertical lines F are the range of possible temperature values embedded in the total space E. The base space K of the fiber bundle is a line because the data points r in E are on a space that is continous in one dimension.

3.2.2 Base Space

The base space K is a representation of the connectivity of the data, specifically whether the points in E are discrete or sampled from a continous space. The same dataset can be expressed with different K.

In our draft implementation of the data as fiber bundle model, we represent K as a simplacial complex.

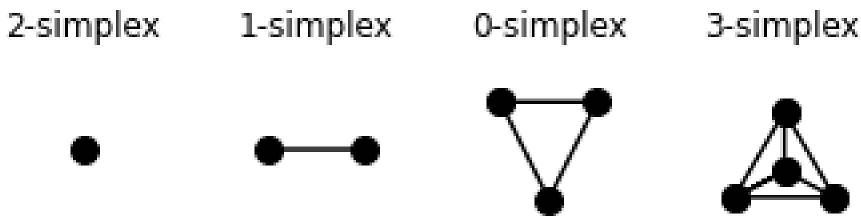


Figure 3: Simplices encode the connectivity of the data, from fully disconnected (0 simplex) observations to all observations are connected to at least 3 other observations. Higher order simplicies are outside the scope of this paper.

One way to represent the topological space K is as a set composed of simplices, such as those shown in figure **??**. Simplices are a way of encoding the connectivity of each observation ($\sigma(k)$) to another:

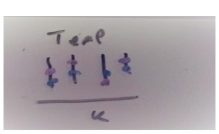
0-simplex discrete observations (inventory records)

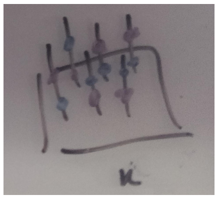
1-simplex 1D continuos data (timeseries)

2-simplex 2D continuos data (map)

3-simplex 3D continuos data (video)

3.2.3 Example





in figure **??**, temperature is the only one data field in r but the K base spaces are different. subfig[1] is a timeseries, so the temperature in r at time t is dependent on the temperature in r_{t-1} and the temperature in r_{t+1} is dependent on r_t; this connectivity is expressed as a one dimensional K where K is the number line. In the case of the map, every temperature in r is dependent on its nearest neighbors on the plane, and one way to express this is by encoding K as a plane. K does not know the time or latitude or longitude of the point as those are metadata variables describing the k rather than the value of k. The mapping $\tau : K \to E$ provides the binding between the key $k \in K$ and the value r in E [13].

3.2.4 Fiber and Sections

We use Spivak's formalization of data base schemas as the basis of our fiber space F [19]. He defines the type specification

$$\pi : U \to DT \tag{10}$$

where DT is the set of data types (as identified by their names) and U is the disjoint set of all possible objects x of all types in DT. This means that for each type $T \in DT$, the preimage $\pi^{-1}(T) \subset U$ is the domain of T, and $x \in \pi^{1}(T) \subset U$ is an object of type T. Spivak then defines a schema (C, σ) of type π, where π is the universe of all types, such that

$$\sigma : C \to DT \tag{11}$$

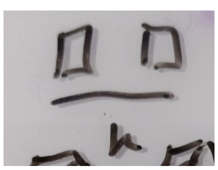
where C is the finite set of names of columuns, which we generalize to data fields in E. The set of all values restricted to the datatypes in DT is U_σ

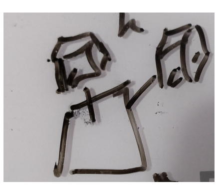
$$\begin{tikzcd} U_\sigma \arrow[r] \arrow[d, "\pi_\sigma"'] & U \arrow[d, "\pi"] \\ C \arrow[r, "\sigma"] & DT \end{tikzcd} \tag{12}$$

The pullback $U_\sigma \coloneqq \sigma^{-1}(U)$ restricts U to the datatypes of the fields in C such that U_σ is the fiber product $U \times_{DT} C$, and the pullback $\pi_\sigma : U_\sigma \to C$ specifies the domain bundle U_σ over C induced by σ. The fiber F is the cartesian product of all sets in the disjoint union U_σ.

For each field $c \in C$, the record function $r : C \to U_\sigma$ returns an object of type $\sigma(c) \in DT$. The set of all records $\Gamma(\sigma)$ is the set of all sections on U_σ. Spivak defines the τ mapping from an index of databases K to records $\Gamma(\sigma)$ as $\tau : K \to \Gamma(\sigma)$. This is equivalent to $\tau : k \to E$ since $F = \Gamma(\sigma)$ and F is the embedding in E on which the records r lie.

3.2.5 Example





The fiber in figure **??** is the space of possible temperature values in degrees celsius, such that $F = [temp_{min}, temp_{max}]$ and is named Temp. In figure **??** time is encoded as a second dimension. This means that the set of possible values F with $C = \{\text{Temp}, \text{Time}\}$:

$$F = [temp_{min}, temp_{max}] \times [time_{min}, time_{max}] \tag{13}$$

and the function τ that retrieves records from F is

$$\tau(k) = (k, (r : \text{Temp} \rightarrow temp, r : \text{Time} \rightarrow time)) \tag{14}$$

$$temp \in [temp_{min}, temp_{max}], time \in [time_{min}, time_{max}]) \tag{15}$$

Since $\tau(k) = (k, r)$, *temp* is bound to a named data field and *sigma* binds *temp* to a temperature data type.

3.3 Prerender Space

We model the prerender space on which lives on ideal version of the visulaization as a fiber bundle (H, S, π, D). H is the predisplay space, with a fiber D dependent on the target physical display and a base space of S.

3.3.1 Base space

K can be considered a subspace of the screen base space S such that $\xi : S \rightarrow K$ is a deformation retraction [15]

$$\begin{tikzcd} E \arrow[d, "\pi"'] & H \arrow[d, "\pi"'] \\ K & S \arrow[l, "\xi"'] \end{tikzcd} \tag{16}$$

that goes from a region $s \in S_k$ to its associated point k, such that when $\xi(s) = k$, $\xi^*\tau(s) = \tau(k)$.

3.3.2 Fiber and Section

A section $\rho : S \rightarrow H$ is a mapping from a region s on a mathematical encoding of the image to a region xy on the screen that the renderer then maps to visual space as defined in D. For a physical screen display, the predisplay space is a trivial fiber bundle $H = \mathbb{R}^7 \times S$ such that ρ is

$$\rho(s) = \{x, y, z, r, g, b, a\} \tag{17}$$

To draw an image, a region, H is inverse mapped into a region $s \in S$ where

$$s = \rho_{XY}^{-1}(xy) \tag{18}$$

such that the rest of the fields in $\mathbb{R}^7$ are then integrated over s to yield the remaining fields:

$$r = \iint\limits_{s} \rho_R(s) ds^2 \tag{19}$$

$$g = \iint\limits_{s} \rho_G(s) ds^2 \tag{20}$$

$$b = \iint\limits_{s} \rho_B(s) ds^2 \tag{21}$$

Here we assume a single opaque 2D image such that the z and *alpha* fields can be omitted.

3.3.3 Example

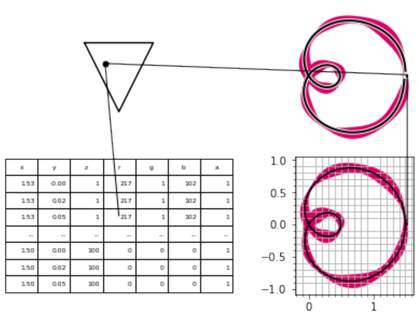


Figure 4

As illustrated in figure 4, words.

3.4 Artist

In this section we will define the artist as a mapping from a sheaf $\mathcal{O}(E)$ to $\mathcal{O}(H)$.

$$A : \mathcal{O}(E) \to \mathcal{O}(H) \tag{22}$$

The artist decomposes to mapping data to visual $\nu : E \to V$, then compositing V pulled back along ξ to ξ^*V to a visual mark in prerender space $Q : \xi^*V \to H$.

$$
\begin{tikzcd}
E \arrow[r, "\nu"] \arrow[dr, "\pi"'] & V \arrow[d, "\pi"] & \xi^*V \arrow[r, "Q"] \arrow[d, "\xi^*\pi"'] & H \arrow[dl, "\pi"] \\
& K & S \arrow[l, "\xi"] &
\end{tikzcd}
\tag{23}
$$

The visual fiber bundle (V, K, π, P) has section $\mu : V \to K$ that resolves to a visual variable [2, 12] in fiber P. The visual transformer ν is a set of functions each targeting a different μ

$$\{\nu_0, \dots, \nu_n\} : \{\tau_0, \dots, \tau_n\} \mapsto \{\mu_0, \dots, \mu_n\} \tag{24}$$

where μ_i are the visual parameters in the assembly function $Q(\mu_0, \dots, \mu_n)(s) = \rho(s)$.

3.4.1 Example: Matplotlib Visual Fiber

For example, for Matplotlib [8], some of the possible types in P are:

$\boldsymbol{\nu_i}$	$\boldsymbol{\mu_i}$	$\boldsymbol{codomain(\nu_i)}$
position	x, y, z, theta, r	$\mathbb{R}$
size	linewidth, markersize	$\mathbb{R}^+$
shape	markerstyle	$\{f_0, \dots, f_n\}$
color	color, facecolor, markerfacecolor, edgecolor	$\mathbb{R}^4$
texture	hatch	$\mathbb{N}^{10}$
	linestyle	$\{f_0, \dots, f_n\} \times (\mathbb{R}, \mathbb{R}^{+n, n\%2=0})$

3.4.2 Visual Channels

$\nu : E \to V$ is an equivariant map such that there is a homomorphism from left monoid actions on E_i to left monoid actions on V_i where i identifies a field in the fiber. E_i and V_i each contain a set of values as defined in F and P respectively. A validly constructed ν is one where the diagram

$$\begin{tikzcd} E_i \arrow[r, "\nu_i"] \arrow[d, "m_e"'] & V_i \arrow[d, "m_v"] \\ E_i \arrow[r, "\nu_i"] & V_i \end{tikzcd} \tag{25}$$

commutes such that $\nu_i(m_e(E_i)) = m_v(\nu_i(E_i))$.

Definition A monoid[11] M is a set that is closed under an associative binary operator $*$ and has an identity element $e \in M$ such that $e * a = a * e = a$ for all $a \in M$. A left monoid action [1, 17] of M is a set X with an action $\bullet$ with the properties:

closure $\bullet : M \times X \to X$,

associativity for all $m, t \in M$ and $x \in X, m \bullet (t \bullet x) = (m \bullet t) \bullet x$

identity for all $x \in X, e \in M, e \bullet x = x$

Example: Partial Order To preserve ordering of elements in E_i, ν must be a monotonic function such that given $e_1, e_2 \in E_i$

$$\text{if } e_1 \leq e_2 \text{ then } \nu(e_1) \leq \nu(e_2) \tag{26}$$

Example: Translation fairly certain I lost the thread here According to Stevens, interval data is a set with general linear group actions [10, 21]. Position is a visual variable that can support translation [2, 9, 12].

For example, here ν is a direct map $\nu(\tau_{i,k}) = \mu i, k$ and the transform function is a multiplicative shift $t(x) = x + x$ applied twice

$$\nu(t(t(\tau_{i,k}))) = t(t(\nu(\tau_{i,k}))) \tag{27}$$

$$\nu(t(2\tau_{i,k})) = t(t_v(\mu_{i,k})) \tag{28}$$

$$\nu(2\tau_{i,k} + 2\tau_{i,k}) = t(2\mu_{i,k}) \tag{29}$$

$$\nu(4\tau_{i,k}) = 2\mu_{i,k} + 2\mu_{i,k} \tag{30}$$

$$\nu(4\tau_{i,k}) = 4\mu_{i,k} \tag{31}$$

$$4\mu_{i,k} = 4\mu_{i,k} \tag{32}$$

$$\tag{33}$$

ν is valid because scaling the data by a factor of 4 causes a scaling of the equivalent visualization by 4, which since the constant factor can be pulled out means they are equivalent.

Example: Invalid ν Given a transform $t(x) = 2 * x$, we construct a ν that always takes data to .5:

$$\begin{tikzcd} E_1 \arrow[r, "\lambda : e \mapsto .5"] \arrow[d, "2e"'] & V_i \arrow[d, "2v"] \\ E_1 \arrow[r, "\lambda"] & V_1 \end{tikzcd} \tag{34}$$

This ν is invalid because the graph does not commute for t:

$$\nu(t(e)) \stackrel{?}{=} t(\nu(e)) \tag{35}$$

$$.5 \stackrel{?}{=} t(.5) \tag{36}$$

$$.5 \neq 2 * .5 \tag{37}$$

To consutruct a valid ν, the diagram must commute for all monoid actions on the sets E_i, V_i.

3.4.3 Assembling Marks

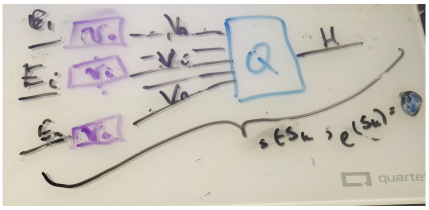


Figure 5: The ν functions convert data E to visual V. Q assembles the different types of visual parameters V_i into a graphic in H. $Q \circ \mu(\xi^{-1}J)$ forms a visual mark by applying Q to a region mapped to connected components $J \subset K$.

The assembly function $Q : \Gamma(V) \to \Gamma(H)$ composites the visual variables in V into an element in H. Given a monoid action on a set in V, there should be a monoid action on the corresponding $Q(\Gamma(V))$. While $\Gamma(V)$ holds for all cases, we can specialize to the bundle V or the sheaf (V) depending on the specific visualization.

Proposition If $\forall g \in M$ and $\forall x_1, x_2 \in \Gamma(V)$ then $Q(x_1) = Q(x_2)$ $impliesQ(g \circ x_1) = Q(g \circ x_2)$; therefore we can define a group action on $Q(\Gamma(V)) = Y$ as $g \circ y = y'$ where $y' = Q(g \circ x)$ with $x \in f^{-1}(y)$

For each region s in the display space H, the mark [2, 5] it belongs to can be found by mapping s back to K via the lookup on S then taking $\xi(s)$ back to a point on $k \in K$ which lies on the path connected component $J \subset K$.

$$H \underset{\rho(\xi^{-1}(J))}{\rightleftarrows} S \underset{\xi^{-1}(J)}{\overset{\xi(s)}{\rightleftarrows}} J_k = \{j \in K \text{ s. t. } \exists\gamma \text{ s.t. } \gamma(0) = k \text{ and } \gamma(1) = j\} \tag{38}$$

where the path[6] γ from k to j is a continous function from the interval [0,1]. To get back to the display space H from J, the inverse image of $J \in S, \xi^{-1}(J)$ is pushed back to S, and then $\rho(\xi^{-1}(J))$ maps it into R^7 such that $Q \circ \xi^* \mu(\xi^{-1}(J))$ generates a graphical mark.

3.4.4 Visual Idioms: Equivalance class of artists

n $O(E)$ of the same type, they output the same type of prerender $O(H)$:

Natural transformation + composition is partial ordering? Back and forth is equivalent

References

[1] *Action in nLab.* https://ncatlab.org/nlab/show/action#actions_of_a_monoid.

[2] Jacques Bertin. "II. The Properties of the Graphic System". English. In: *Semiology of Graphics.* Redlands, Calif.: ESRI Press, 2011. ISBN: 978-1-58948-261-6 1-58948-261-1.

[3] D. M. Butler and M. H. Pendley. "A Visualization Model Based on the Mathematics of Fiber Bundles". en. In: *Computers in Physics* 3.5 (1989), p. 45. ISSN: 08941866. DOI: 10.1063/1.168345.

[4] David M. Butler and Steve Bryson. "Vector-Bundle Classes Form Powerful Tool for Scientific Visualization". en. In: *Computers in Physics* 6.6 (1992), p. 576. ISSN: 08941866. DOI: 10.1063/1.4823118.

[5] Sheelagh Carpendale. *Visual Representation from Semiology of Graphics by J. Bertin.* en.

[6] "Connected Space". en. In: *Wikipedia* (Dec. 2020).

[7] "Fiber Bundle". en. In: *Wikipedia* (May 2020).

[8] J. D. Hunter. "Matplotlib: A 2D Graphics Environment". In: *Computing in Science Engineering* 9.3 (May 2007), pp. 90–95. ISSN: 1558-366X. DOI: 10.1109/MCSE.2007.55.

[9] John Krygier and Denis Wood. *Making Maps: A Visual Guide to Map Design for GIS.* English. 1 edition. New York: The Guilford Press, Aug. 2005. ISBN: 978-1-59385-200-9.

[10] W A Lea. "A Formalization of Measurement Scale Forms". en. In: (), p. 44.

[11] "Monoid". en. In: *Wikipedia* (Jan. 2021).

[12] T Munzner. "Marks and Channels". In: *Visualization Analysis and Design*, pp. 94–114.

[13] Tamara Munzner. "Ch 2: Data Abstraction". In: *CPSC547: Information Visualization, Fall 2015-2016* ().

[14] "Quotient Space (Topology)". en. In: *Wikipedia* (Nov. 2020).

[15] "Retraction (Topology)". en. In: *Wikipedia* (July 2020).

[16] Todd Rowland. *Fiber Bundle*. en. https://mathworld.wolfram.com/FiberBundle.html. Text.

[17] "Semigroup Action". en. In: *Wikipedia* (Jan. 2021).

[18] E.H. Spanier. *Algebraic Topology*. McGraw-Hill Series in Higher Mathematics. Springer, 1989. ISBN: 978-0-387-94426-5.

[19] David I Spivak. "SIMPLICIAL DATABASES". en. In: (), p. 35.

[20] "Stalk (Sheaf)". en. In: *Wikipedia* (Oct. 2019).

[21] S. S. Stevens. "On the Theory of Scales of Measurement". In: *Science* 103.2684 (1946), pp. 677–680. ISSN: 00368075, 10959203.