Visualization Institute of the University of Stuttgart

University of Stuttgart
Allmandring 19
D–70569 Stuttgart

Master Thesis Nr. 1025004

Visual Exploration of Workload Performance Data

Krishna Damarla

Course of Study:	Information Technology
Examiner:	Prof. Dr. Daniel Weiskopf
Supervisor:	Dipl.-Inf. Christoph Schulz, Dipl.-Inf. Nils Rodrigues, Dipl.-Inf. Andreas Henicke (IBM)
Commenced:	December 1, 2016
Completed:	June 1, 2017
CR-Classification:	I.3, D.4.8, H.5

Abstract

Multiple ways exist to support the decision making process of Work Load Management (WLM) team. Currently, the WLM team at IBM Böblingen R&D labs uses Excel charts for visual exploration of different records. Though this solution covers basic visualizations, there is need for a solution that is dynamic, scalable and suitable for visualization of various multidimensional records to get better prospects of information from the system generated reports.

The goal of this thesis is to build a tool that makes inspection much easier for human cognition, faster with interactions by dynamically adapting to various SMF reports and scale through various attributes irrespective of changing attribute names, number of attributes, and number of observations from one report to the another.

At first, we discuss the basic background for this project: Information visualization and visual analytics. After reviewing various record formats and report evaluations, we explore various visualization techniques and briefly introduce the software design process involved. Later, we go through the implementation details and used technologies for achieving the desired goals and carry out visual analysis. Finally, we perform a user case study to show the usefulness and applicability of the developed tool.

We formally implement eight out of many visualization techniques possible: Line Plot, Multiline Plot, Scatter Plot, Dot Plot, Density Plot, Correlation Heatmap, Scatter Plot Matrix, Parallel Coordinates over various WLM and RMF (Resource Measurement Facility) reports. These plots enable user interaction and visual encodings to support visual analysis. In addition, computational analysis like clustering, correlation are also supported for some of these visualization techniques.

Contents

List of Figures

List of Tables

Listings

1 Introduction

This master thesis is a joint project with the visualization research center of university of stuttgart and IBM Systems group of Böblingen R&D labs. In this chapter, we briefly describe what z/OS is, what components we will be dealing with and their influence on the system, goal and timeline of this project.

1.1 Motivation

IBM System z or zSeries family is a part of IBM mainframe computers. Z systems evolved over time from system 360 to current z13 series. Despite continual changes, mainframe computers are compatible and yet continue to run applications written in the 70s. [EKO+12]

Today, applications of mainframe computers range from everyday ATM transactions to flight bookings. One of the critical component responsible for this is z/OS operating system. z/OS operating system is a share-everything run time environment which uses special software and hardware to access and control use of resources, ensuring that there is little under utilization of its components.

One of the strengths of the z/OS is the ability to run multiple workloads at the same time. The function that makes this possible is dynamic workload management implemented in the workload manager component of the z/OS operating system.

WLM manages system resources such as storage and processors. Also, it manages the performance characteristics of transactions, processing of workloads in the system according to the company's business goals. WLM plays an important role in advising the system for optimized work distribution which has direct influence on the performance of system.

Hence, understanding various factors of WLM, how they are changing over time and patterns between them through visual means is crucial for making decisions that have direct impact on system

1.2 System Environment

There are many ways to control the system depending on point of emphasis. Through HMC (Hardware Management console) which is located in the operator area or through specialized microprocessors for internal control functions which are located inside system but can be used by operators. This start of system is also called IPL (Initial Program Load).

A sophisticated mainframe can run more than one operating system at same time. The system typically comprised of different processor units like CP (Central Processor), IFL (Integrated Facility for Linux) and so on. The CP in current mainframes contains: z/OS (developed from MVS, OS/390), z/VM, z/VSE, z/TPF (successor of Airlines Control Program), Linux on z Systems. In this project, we concentrate on widely used mainframe operating system z/OS.

1.2.1 z/OS

z/OS is commonly referred to as system software or BCP (Base control Program). The structure of the z/OS system as stated by Robert in High Availability and scalability of mainframe environments in system z [Vau13] is as shown in Figure 1.1. [EKO+12]

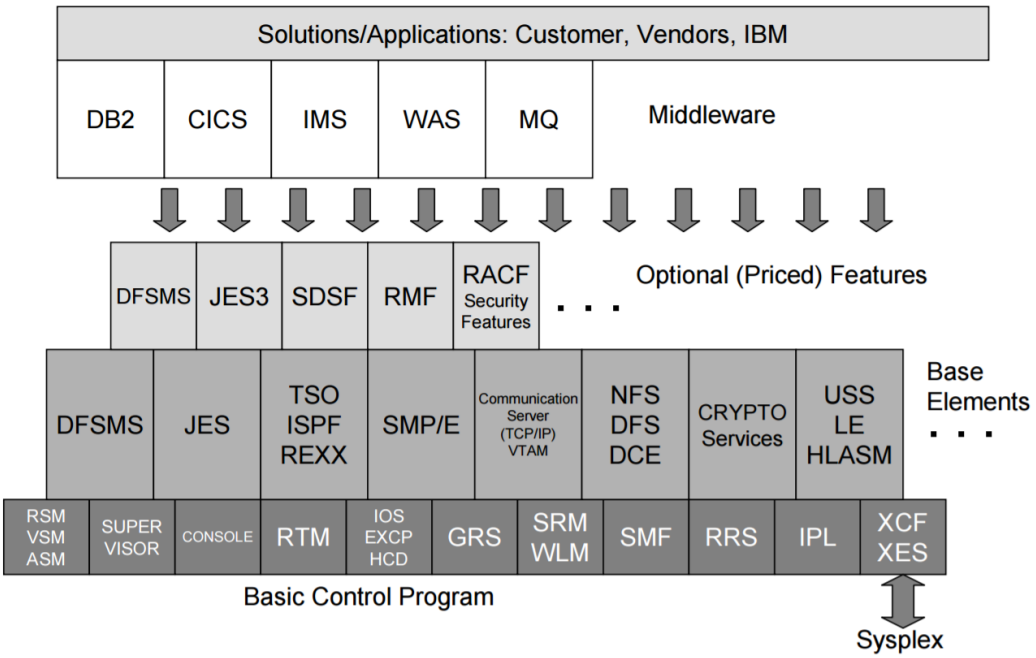

Figure 1.1: Structure of z/OS System [Vau13]

An extensive set of unique attributes and system facilities makes z/OS well suited for processing large, complex workloads that include long running applications which update millions of records in databases, batch processing, online applications which serve thousands of users concurrently. z/OS virtual storage is responsible to handle huge number of users concurrently while processing the large workloads.

There are various operating system functions listed in Figure **??** like supervisor/dispatcher, console services, recovery, storage management, unix system services, IPL, sysplex communication. Among them, we mainly concentrate on the WLM and RMF components.

1.2.2 Work Load Management (WLM)

WLM is an attempt to optimize throughput to satisfy end user goals for the work on single system across a cluster and across a CEC. Note the terminology overlap of zSeries machine in various books. Sometimes it is referred to as Processor/CPU/CEC/System.

WLM has three main objectives:

1. Goal Achievement: To automatically assign system complex (sysplex) resources to workloads based on their importance.

2. Throughput: Achieving optimal use of system resources from system point of view.

3. Response and turn around time: Achieving optimal use of system resources from the individual address space point of view. An address space is the range of virtual addresses a operating system assigns to user.

How WLM works can be simplified as below:

1. Identify work running in system (Middleware & OS tells WLM when a new unit of work enters or leaves system)

2. Measure the time it runs

3. Find the contention points. (Contention occurs when work wants to use system resources like CPU's, I/O devices, Storage or Software constructs like Processes or Address Spaces. WLM monitors these resources to understand how many resources work requires)

The important OLTP (Online Transaction Processing) transactions which are critical to run gets the highest priority and gets the resources allocated whereas heavy batch workloads gets low priorities and are shifted to the night times. For more details, Refer to "System Programmer's Guide to Workload Manager"[Red07]

Using goal mode system operation, WLM and System Resource Manager (SRM) match resources to meet the goals by constantly monitoring and adapting the system. WLM provides solution for managing workload distribution, workload balancing, and distributing resources to competing workloads.

The supervision of goal achievement is done with simple metric called Performance Index (PI). If PI=1, the goal is achieved; if the value is smaller or greater than 1, the goal is missed.

Response Time Goal:

$$PI\_RT = \frac{ActualAchievedResponseTime}{ResponseTimeGoal} \tag{1.1}$$

Execution Velocity Goal:

$$PI\_EV = \frac{ExecutionVelocityGoal}{ActualAchievedExecutionVelocity} \tag{1.2}$$

PI is calculated as shown in equations 1.1 and 1.2 for each service class and is a deciding factor to understand whether goals are met or not from WLM's perspective. For every ten seconds, WLM summarizes system information along with CPU service, transaction end times. Here, WLM knows the behavior of workloads and system. This historical data is stored and later visualized to learn the relationship between various factors within the system. [Red07]

WLM on its own can do really little to understand what is happening on system. Therefore, an important aspect is its coordination with the main software components that execute on the system. z/OS components like transaction managers, database managers communicate with WLM to change status for particular address space. In this project, we shall see about RMF records to better understand the WLM goals[Vau13]

1.2.3 Resource Measurement Facility (RMF)

RMF is an IBM licensed program used to measure various aspects of system performance. Different RMF modules provide instantaneous data, batch-type reports, time sharing option oriented reports, long-term reporting, long-term statistical gathering and so on.

RMF also accesses the hardware counters like queuing time for each i/o device, amount of activity per device etc., and includes all these information in its reports. In order to report system performance, RMF needs to first gather information about system and its activity.

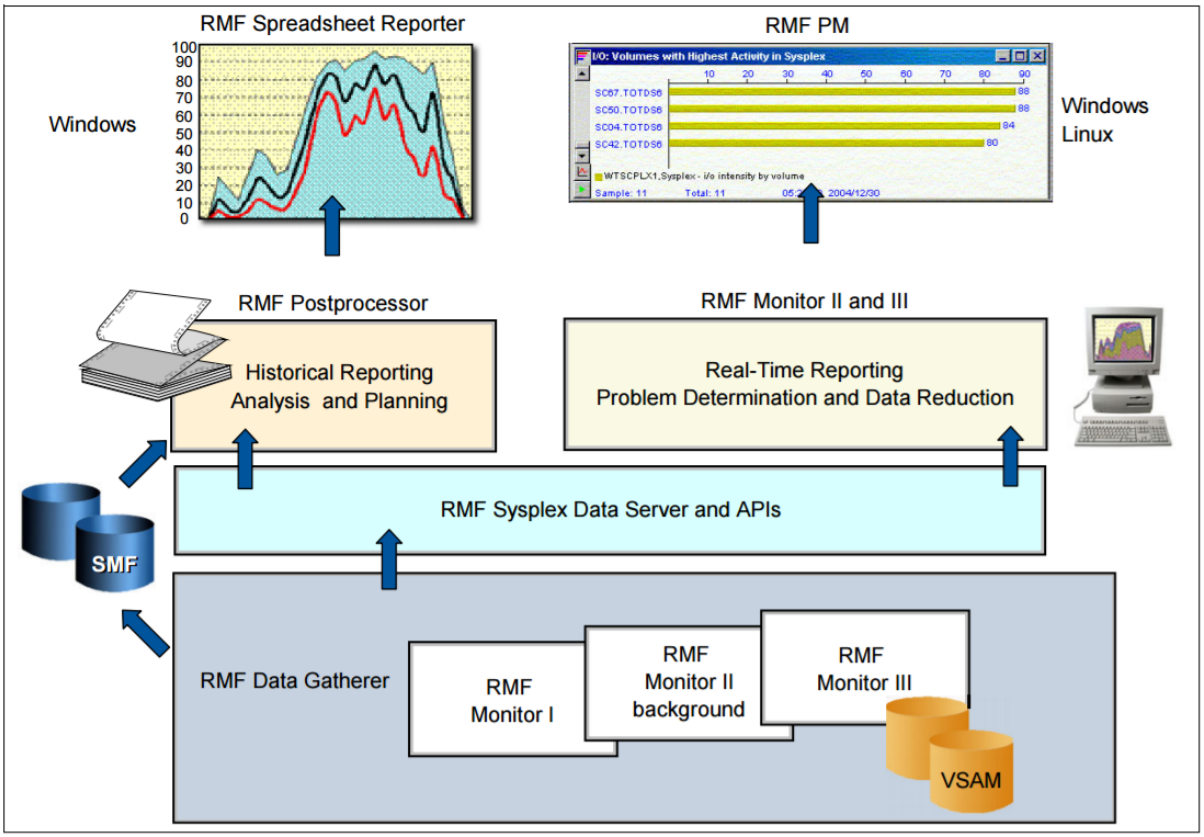

Figure 1.2: A graphical overview of functions and interfaces provided by RMF in z/OS [Red05]

As we can see in Figure 1.2, Monitor I and Monitor III gather long term data that is collected in records in 15 or 30 minute intervals, reports created by snapshot monitoring with Monitor II are generated every time user click enter, short term data collection is done with Monitor III and the data collected is defined in seconds or minutes. For monitor III, default time interval for gathering or reporting data is 100seconds.

RMF stores z/OS data in two types of records: SMF records (All three monitors write SMF records) and VSAM records (Only monitor III writes VSAM records)

Workload Activity reports are contained in SMFs 72 records. So, in this project, we concentrate on RMF's 72 records and in particular subtype three which is written for each service class/report class in the active policy along with WLM's 99 records. The details about these records will be presented in section 2.2.

1.3 Problem Statement

Various factors are taken into consideration when looking at the performance of a system. For example, user response time, throughput, device utilization, number of users supported, reliability, system capacity and resources that affect above aspects like applications, storage, CPU, I/O, networking, paging.

These factors are monitored and stored into various records for evaluation, thereby enabling business goal oriented decision-making.

Currently, visual exploration is supported by excel spreadsheet tool. The visual models used for exploration with this tool are not suited for visualizing multivariate data. Also, users need to have more domain knowledge and setups to run different executable files for visualizing specific set of records.

The aim of this thesis is to build a tool that solves the above problems through interactive visual models. It dynamically adapts to various records, decreases the setups and domain knowledge required. Further, it scales to multiple dimensions irrespective of change in attributes from different records or change in observations with respect to the samples monitoring time interval.

We also make sure that visual models are easier for human cognition with interactive visual analysis. This will significantly improve the WLM decision-making capacity with respect to the system performance.

1.4 Timeline

The gantt chart depicted in Figure 1.3 gives us the brief outline for this project. The project started in December after attending the prerequisite classes about information visualization and visual analytics at university of stuttgart.

At first, the data review phase involved communicating with industry experts, gathering requirements, understanding about various records and meta data of the records. Later, we proceeded to requirements engineering which involves necessary installations required for the software development, literature review and initial prototypes.

Further, we continued with design decisions of various visual models and implementation of the modules before integrating them with the user interface.

Finally, we performed the user case study and gathered suggestions, exploration views from various WLM and RMF users.

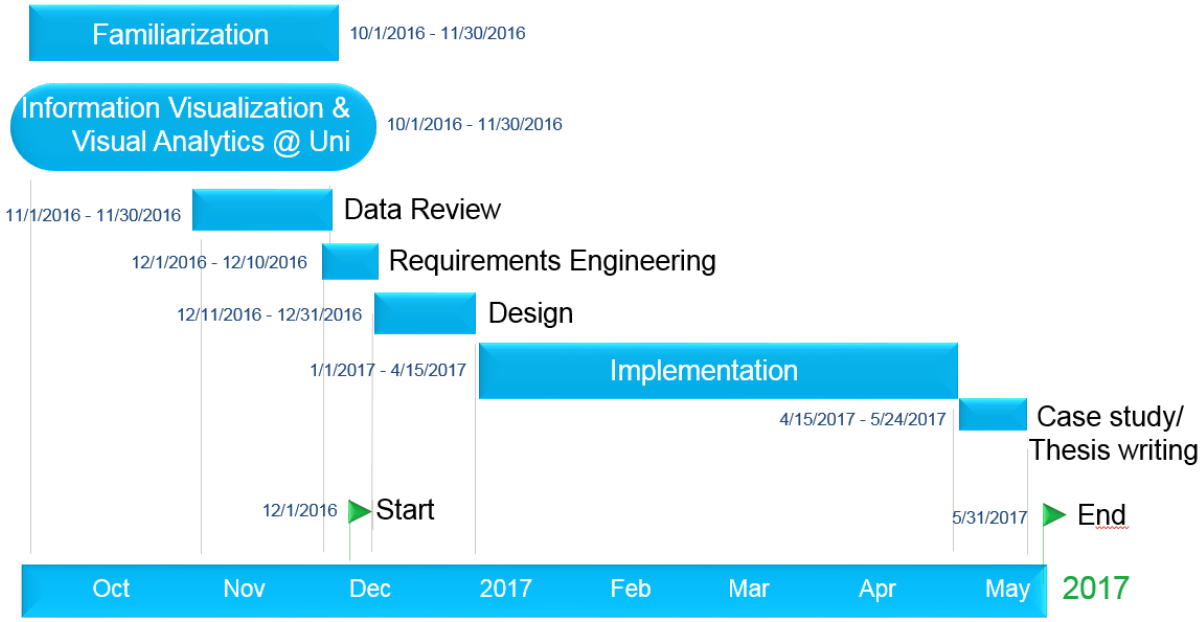

Figure 1.3: Project Timeline

2 Background

2.1 Information Visualization and Visual Analytics

"Graphics is the visual means of resolving logical problems." - Bertin [Ber81]

"Information Visualization (InfoVis) is the process of mapping abstract data into a graphical representation and thereby to enable human brain to understand and interact with data sets. Visual Analytics focuses on analytical reasoning through interactive visual interfaces." [Wor11]

2.1.1 InfoVis Reference Model

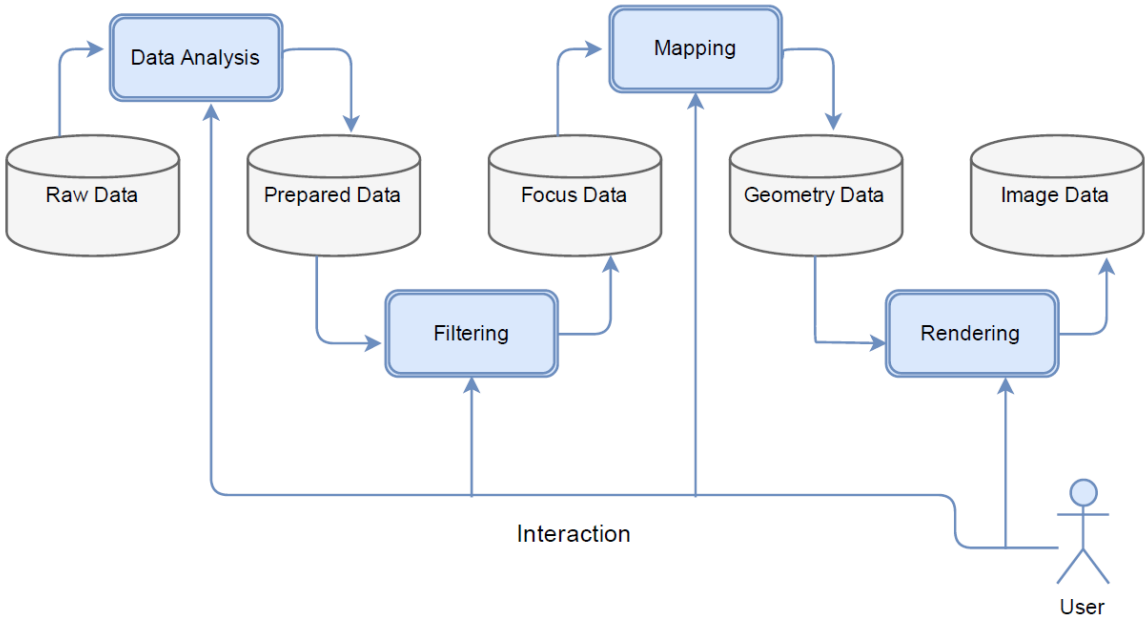

Figure 2.1: Information Visualization Pipeline for Visual Analytics [wike][AMST11]

Below steps explains information visualization pipeline mentioned in Figure 2.1 for generating visual representations from the abstract data.

1. At first, raw data is processed/prepared for visualization. For instance, by interpolating missing data, correcting wrong measurements and so on. We see this process in detail in section 2.2.

2. After data got processed, we select portions of data to be visualized. This selection can be done by user control or by automatically pre-selecting data that is of interest to user through algorithmic methods.

3. Later, focused data is mapped to geometric primitives like square, line, circle etc., and with different visual encoding like position, color, size. This usually involves scaling, normalizing, sub-sampling, interpolating, and shifting data prior to mapping. As this is the heart of the information visualization pipeline, we see this step in detail in later subsections.[WGK10]

4. Finally, the geometric data is transformed to the image data.

The important component of visualization pipeline is user interaction through data transformations (filter, aggregate, extract), visual mappings and view transformations (changing visual perspective through zoom, pan, linking & brushing). In earlier days, visualizations were static. But with modern visualization, user can dynamically interact with all aforementioned stages. Typically, the user interface consists of components which requests data, monitors, analyzes and performs computations over data. [AMST11]

2.1.2 Types of Attributes

Attributes within a data record can be broadly classified as Categorical/Nominal or Ordered.

a) Categorical Attributes

Categorical/Nominal attributes take non-numeric values such as names of CPU's or physical addresses and so on. These attributes does not have implicit ordering. They only distinguish whether two items are same or different as shown in the Figure 2.2.

Figure 2.2: An example depicting categorical type of attributes [MM15]

b) Ordered Attributes

Ordered attributes have implicit ordering as opposed to unordered nominal attributes.This is further subdivided to two categories as show in the Figure 2.3[MM15]

1. Ordinal Attributes: Ordinal data such as shirt size (small, large) or ranked data (high, medium, low). Though not suitable for arithmetic calculations, it clearly conveys a well-defined ordering.

2. Quantitative Attributes: These attributes takes only numeric values such as binary data (0 or 1) or discrete data (3, 6, 9) or continuous data (real values within interval like [10,15]). Both integers and real numbers are of quantitative type.

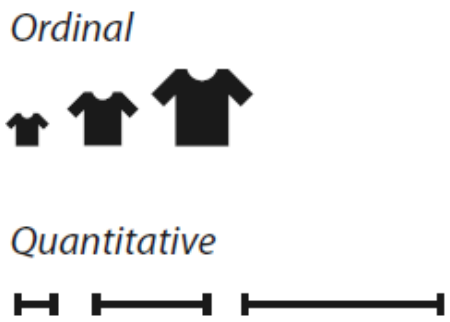

Figure 2.3: An example of ordered type of attributes [MM15]

Ordered data can be further categorized as sequential/diverging/cyclic as shown in the Figure 2.4 :

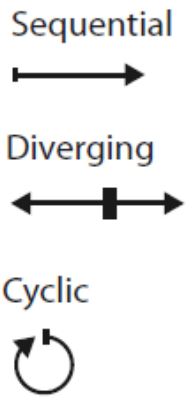

Figure 2.4: An example of ordering directions [MM15]

1. Sequential: This data range from minimum value to a maximum value or vice versa. For instance, mountain height measured from sea level to tip.

2. Diverging: Data pointing in opposite directions with two sequences that meet at common zero point.

3. Cyclic: Data values wrap back to starting point rather than increase or decrease indefinitely. For instance, day of week or hour of day.

2.1.3 Types of Visual Encoding

We map the categorical or ordered attributes with respect to color/size/shape/shape and many others as shown in the Figures 2.5 and 2.6. Color: Change of color with respect to hue, saturation, luminance for a given value.

Size: Change of size with respect to length, area and repetition of values.

Shape: Assigning different shapes to different categories of attribute

Motion(direction, rate, frequency): Best suits for highlighting selected data

and many others geometric primitives like angle, curvature, spatial position, blur and so on.

These features are preattentively processed by human brain[Hof00]. Initiated by Bertin[Ber81] and extended by others [Mac86], [WWR+05],[BAB+93], the research of these features continues and provides valuable insights for foundations of visualizations.

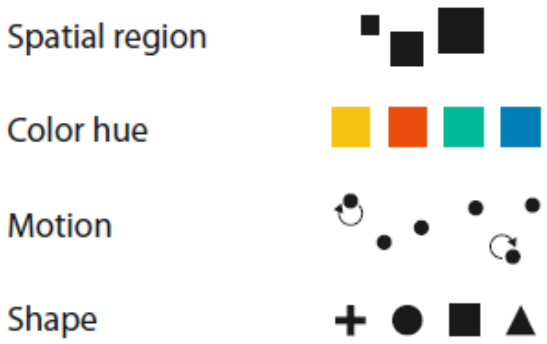

Figure 2.5: Visual encoding channels for categorical attributes [MM15]

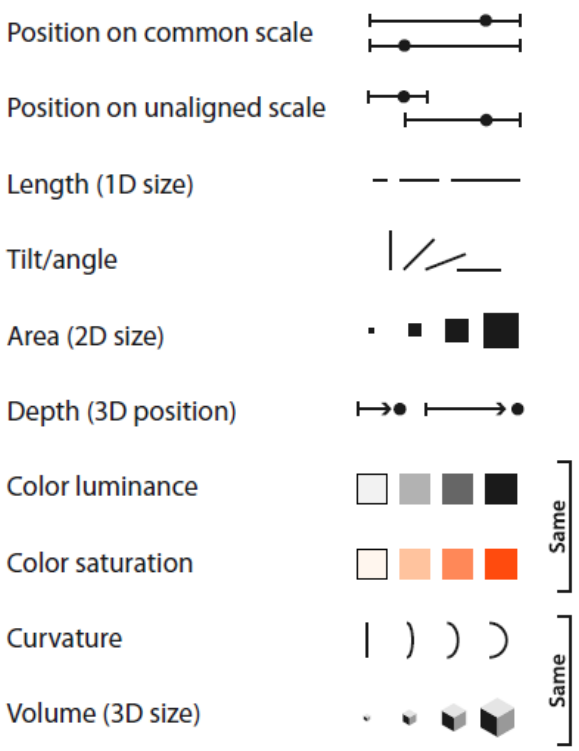

Figure 2.6: Visual encoding channels for ordered attributes [MM15]

2.1.4 Scaling

Another way of categorizing attributes is by the mathematical concept of scaling. Scale is an important factor to examine for each specific visualization design as each attribute has a appropriate scale associated with it. According to Stevens[Ste46], there are four scales of measurement:

1. Nominal scale: This is for naming items or for categorical data. There is no specific ordered sequence defined. Sometimes, this scale is also used for numbers. For example, the number of a bus line, clearly defines a nominal value which specifies the route on which the bus travels.

2. Ordinal scale: This scale is used for ordering items in a sequence. It is often the case, where some items come strictly before or after certain items. We use it for example, when we create a ranking of group of items like number of resources. The position of item is clearly an ordinal quality.

3. Interval scale: Here, the differences between items are defined (no ratio). For instance, time-series data.

4. Ratio scale: With this scale, comes the power of expressing real numbers. For instance, Comparing weight of one object with another object or height of items. The ratio scale implies a meaningful zero value as reference.

2.1.5 Normalization

It is the process of transforming an attribute so that it fits within the specified range. It is an important factor especially for visualization.

Normalization maps the values of a attribute in the given record to a value between minimum and maximum.[WGK10]

For instance, if an attribute X has minimum and maximum values, normalizing each value of X is done with respect to,

$$X_{Normalize} = \frac{X_{Value} - X_{Minimum}}{X_{Maximum} - X_{Minimum}} \tag{2.1}$$

It helps to reduce the scales to the user screen display for visualization. There are different kinds of normalization's available: linear/logarithmic/square root. In practice, attributes contain different range values. We need to normalize this data to be compatible for visualization.

2.1.6 Discrete and Continuous Visualizations

Tory [TM02] established a model for the taxonomy in the context of visualizations (see Figure 2.7). According to that, visualization algorithms are broadly classified to their data model as discrete or continuous.

InfoVis mostly deals with discrete models and Scientific Visualization (SciVis) deals mostly with continuous data models. However, this analogy does not apply for every data model. The major difference between them is, SciVis is used for clarifying well-known phenomena and InfoVis is used for searching for interesting phenomena.[Nag06]

InfoVis taxonomies deals with multidimensional databases, text, graphs, trees. In addition, some visualizations are classified by representation/display style. SciVis is mainly classified by number of independent/dependent variables and whether data is scalar, vector, tensor, or multivariate.

Discrete models are classified according to whether data points are connected or disconnected. Later, we further separate visualization techniques according to the number of dimensions the visualization supports. For instance, one, two, three, or N dimensions.

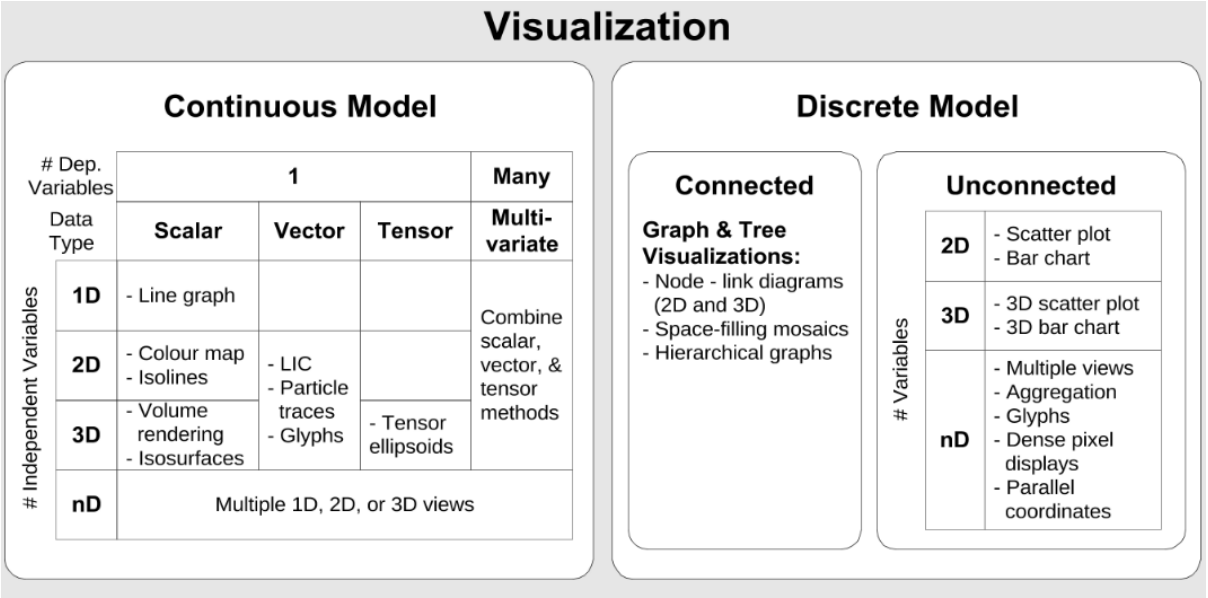

Figure 2.7: Visualization Taxonomy [TM02]

2.2 Reviewing and Evaluating SMF Records

The data about various software components of the system is collected and stored into the SMF records. As specified in section 1.2.2 and section 1.2.3, we are concentrating on the software components WLM and RMF of the system. Hence, we consider the SMF records that contains data about these two components.

SMF records range from 0 t0 120. Out of those, we are concentrating on SMF records 72 and 99. Record type 72 belongs to the RMF component. Record type 99 belongs to the WLM component.

SMF records of type 72 have five subtypes. Out of which two are deprecated. Subtype 3 till 5 gives data about workload activity, storage data and serialization delay. SMF records 99 are divided into 14 subtypes. Subtype 1 contains system level data, subtype 2 for service classes, 3 for service class period plot data and so on. For more details, Refer to "IBM Knowledge Center: z/OS MVS"[Boo]

2.2.1 Data Conversion

The raw binary data of SMF records is converted to CSV format using REXX program in a batch environment. Different REXX programs are used for converting various records. In this project, we are using CSV datasets generated using such REXX programs written internally by developers for processing respective records[wikc]. Figure 4.7 is a sample snippet of REXX script for processing SMF 19 records.

```
/*REXX---------------------------------------------------------------
 DASD Report from SMF19 Records
 ---------------------------------------------------------------------*/
call initialize
call process_smf_data
exit 0
/*---------------------------------------------------------------------
 ---------------------------------------------------------------------*/
initialize: procedure expose __.
  __. = ''
  __.0border = copies('-',70)
  return
/*---------------------------------------------------------------------
 ---------------------------------------------------------------------*/
process_smf_data: procedure expose __.
  "CALL 'SYS1.LINKLIB(IFASMFDP)'"
  "REPRO IFILE(DUMPOUT) OFILE(WORKSMF)"
  "EXECIO * DISKR  WORKSMF ( FINIS STEM SMF."
  drop DASD_OUT.
  count = 1
  j = 1
  say date() time() "DASD Report => read " smf.0 "input records"
  /*-------------------------------------------------------------------
   -------------------------------------------------------------------*/
  do i = 1 to smf.0
     line      = smf.i
    /*-----------------------------------------------------------------
     The standard SMF record header with subtypes . . .
     We're ignoring SMFxLEN (Offset 00) and SMFxSEG (Offset 02) since
     the IDCAMS PRINT utility doesn't include the record length and
     segment descriptor fields.
     -----------------------------------------------------------------*/
    SMFxFLG = binary_b(line,offset(4),1)
    rectype = binary_d(line,offset(5),1)  /* SMFxRTY: Record type    */
    say 'Line:' right(i,6,'0') 'Record Type:' rectype
```

Figure 2.8: Sample snippet of REXX program to produce CSV files [wikc]

2.2.2 Data Categorization

2.2.2.1 Time-Varying Data

The sample that is taken from CSV dataset can be categorized as flat table with a row representing the observation and each column representing a different attribute.

Each observation is taken depending on the interval set for corresponding RMF/WLM records.

Date	Time	Time-Selec	Reporting	LPAR Num	LPAR Nam	System Na
11/17/2015	0:00:00	0	OS07	1	OS07	OS07
11/17/2015	0:00:00	0	OS07	4	OSP1	OSP1
11/17/2015	0:00:00	0	OS07	5	OST1	OST1
11/17/2015	0:00:00	0	OS07		Attribute	OS05
11/17/2015	0:00:00	0	OS07	7	OS06	OS06
11/17/2015	0:00:00	0	OS07	8	OS16	OS16
11/17/2015	0:00:00	0	OS07	9	OS23	OS23
11/17/20	Observation/sample		OS07	11	OS25	OS25
11/17/2015	0:00:00	0	OS07	13	OS35	OS35
11/17/2015	0:00:00	0	OS07	1	Cell value	YSC
11/17/2015	0:00:00	0	OS07	16	COMP	COMP
11/17/2015	0:00:00	0	OS07	19	CBUQ	CBUQ
11/17/2015	0:00:00	0	OS07	4	PK1	PK1
11/17/2015	0:00:00	0	OS07	5	SK1	SK1
11/17/2015	0:00:00	0	OS07	6	SPR1	SPR1
11/17/2015	0:00:00	0	OS07	8	SYS2	T005

Figure 2.9: A Sample Table of LFLTOSO7 Dataset

The CSV file specified in Figure 2.9 consists of attributes: Date, Time, Time-select and so on with multiple rows.

The temporal attributes like date and time are key dimensions which classify the dataset as time-varying. Data about time is difficult to handle as it isn't bound to a single scale. For instance, the scale could range from seconds, hours to weeks.

2.2.2.2 Attribute Type Categorization

The Figure 2.10 is from the record S991DATA. The attributes within this dataset are broadly classified according to section 2.1.2. For instance, column A contains sequential data, column B and C contains date and time data, column D contains categorical data and so on.

A	B	C	D	E	F	G	H	I	J	K
Intv	MM/DD/YY	HH:MM:SS	SID	Tot%	CPU%	IFA%	#CP	#IF	IFANRM	ImpSyst
1	12/8/2015	11:13:18	OS25	65.75	99	0	10	0	256	511778
2	12/8/2015	11:13:28	OS25	65.25	98.81	0	10	0	256	496769
3	12/8/2015	11:13:38	OS25	65.87	98.87	0	10	0	256	531999
4	12/8/2015	11:13:48	OS25	65.62	98.75	0	10	0	256	516233
5	12/8/2015	11:13:58	OS25	66.25	98.68	0	10	0	256	540195
6	12/8/2015	11:14:08	OS25	67.43	97.18	0	10	0	256	584093
7	12/8/2015	11:14:18	OS25	68.56	98.06	0	10	0	256	493965
8	12/8/2015	11:14:28	OS25	68.43	98.93	0	10	0	256	489819
9	12/8/2015	11:14:38	OS25	70.31	99.75	0	10	0	256	545928
10	12/8/2015	11:14:48	OS25	69.31	99.37	0	10	0	256	493859
11	12/8/2015	11:14:58	OS25	69	99.68	0	10	0	256	523904
12	12/8/2015	11:15:08	OS25	67.75	99.75	0	10	0	256	646084
13	12/8/2015	11:15:18	OS25	66.06	99.81	0	10	0	256	518407
14	12/8/2015	11:15:28	OS25	65.81	99.87	0	10	0	256	517728
15	12/8/2015	11:15:38	OS25	67.62	99.93	0	10	0	256	544163
16	12/8/2015	11:15:48	OS25	66.93	99.93	0	10	0	256	483307
17	12/8/2015	11:15:58	OS25	68.87	100	0	10	0	256	510950
18	12/8/2015	11:16:08	OS25	67.81	100	0	10	0	256	538196
19	12/8/2015	11:16:18	OS25	67.25	100	0	10	0	256	446149
20	12/8/2015	11:16:28	OS25	66.75	99.87	0	10	0	256	533033
21	12/8/2015	11:16:38	OS25	67.18	99.81	0	10	0	256	564744
22	12/8/2015	11:16:48	OS25	66.43	99.81	0	10	0	256	459222
23	12/8/2015	11:16:58	OS25	67.31	99.12	0	10	0	256	513930
24	12/8/2015	11:17:08	OS25	67.06	99	0	10	0	256	549954
25	12/8/2015	11:17:18	OS25	65.37	98.62	0	10	0	256	428827
26	12/8/2015	11:17:28	OS25	65.87	99.06	0	10	0	256	497543
27	12/8/2015	11:17:38	OS25	67.12	99	0	10	0	256	513470

Figure 2.10: The S991 dataset with attributes of sequential, date-time & categorical data

2.2.3 Analytical Abstraction

Analytical Abstraction means data about data. It is also known as meta data. All the CSV files that were generated after data conversion follow a particular pattern that clearly distinguishes them from record type to record type and in particular to which subset that record belongs to. For instance, the S991DATA file belongs to record type 99 of subtype 1.

There are some instances when the file name does not match with the semantics. In that case, understanding the attributes inside the file and the timing interval it follows is necessary for identification. Table 2.1 describes attributes of S99 record for subtype 8.

Attribute	Description
Intv	Unique identifier for each record
MM/DD/YY	Date of the samples, format is Month-day-year
HH:MM:SS	Time in format, Hours-minutes-seconds
SID	Unique identifier of the system on which this record is created.
ImageNam	Image system name
OFR	Overflow record matched
NOD	No DIAG (Diagnostic information)
NOL	LPAR CPU management not enabled
ERR	Error in DIAG
NOI	Image entry not written to DIAG
LNX	Linux image
IID	Image id of the partition
SLT	System slot number
UTIL	Average CPU utilization
TCPU	Total CPU weight
ACPU	Active CPU's
TOTWGT	Total weight
INIWGT	Image Initial weight
CURWGT	Image current weight
MINWGT	Image minimum weight
MAXWGT	Image maximum time
PROTIME	Total processor time available
SRVUNITS	Service units
SOFTCAP	Soft capping MSU's
PRCWGT	Current pricing management weight

Table 2.1: S998DATA Attributes Description

Here, we are providing the description of one sample SMF record. However, the attribute description of other records can be found from appendix and from respective links.[Boo]

2.2.4 Evaluation

After a brief review of various records, the questions that arise are in general:

How data is changing over time ? (2.2)

What is common over various records ? (2.3)

How one factor is related to the other ? (2.4)

How frequent are the observations ? (2.5)

Why some factors change ? (2.6)

What factors are important for decisions ? (2.7)

Are there any patterns ? (2.8)

Do factors fall with in expected interval ? (2.9)

Are there clusters in the data ? (2.10)

What conclusions can be drawn from data ? (2.11)

And many others. These questions cannot be answered just by reviewing data from record files. However, simply plotting the same data provides a clear insight.

Thus, we explore in detail various visualization algorithms which fit through this time-varying data in later sections. Finally, we find answers for most of our questions by the end of this thesis report.

3 Related Work

3.1 Literature Review

Depending on the data to be explored, various visualization techniques are proposed and those which suit for the data at hand are used for visual exploration. Some visualization techniques are suitable for low-dimensional data and some for multidimensional data sets. [Kei01]

These visualization techniques are mainly classified by three factors: Data to be visualized, the technique itself and the interaction method.

Interaction techniques like filtering, highlighting, zooming, details on demand, focus and context, linking and brushing and many others allows users to interact with the visualization and make dynamic changes to the visualization according to the exploration objectives. For visualizing particular dataset, any of the following visualization techniques can be used in conjunction with any of the interaction methods.[WGK10]

3.1.1 Line Plot

Features	Description
Horizontal Axis	Time Attribute
Vertical Axis	Quantitative or Ordered Attribute
Encoding	Expressing values of both the attributes with a line connecting the data points over scaled spatial positions
Interactions	Axis and label change
Functionality	Evolution over time and finding outliers

Table 3.1: Features of Line Plot

A Line Plot is used to visualize time series data by taking time on horizontal axis. Later, the data points are plotted with respect to the vertical axis positions as shown in the Figure 3.1. The features of a line plot are depicted in Table 3.1.

MM/DD/YY	HH:MM:SS	SID	Tot%
12/8/2015	11:13:18	OS25	65.75
12/8/2015	11:13:28	OS25	65.25
12/8/2015	11:13:38	OS25	65.87
12/8/2015	11:13:48	OS25	65.62
12/8/2015	11:13:58	OS25	66.25
12/8/2015	11:14:08	OS25	67.43
12/8/2015	11:14:18	OS25	68.56
12/8/2015	11:14:28	OS25	68.43
12/8/2015	11:14:38	OS25	70.31
12/8/2015	11:14:48	OS25	69.31
12/8/2015	11:14:58	OS25	69
12/8/2015	11:15:08	OS25	67.75
12/8/2015	11:15:18	OS25	66.06
12/8/2015	11:15:28	OS25	65.81
12/8/2015	11:15:38	OS25	67.62
12/8/2015	11:15:48	OS25	66.93
12/8/2015	11:15:58	OS25	68.87
12/8/2015	11:16:08	OS25	67.81
12/8/2015	11:16:18	OS25	67.25
12/8/2015	11:16:28	OS25	66.75
12/8/2015	11:16:38	OS25	67.18
12/8/2015	11:16:48	OS25	66.43
12/8/2015	11:16:58	OS25	67.31
12/8/2015	11:17:08	OS25	67.06
12/8/2015	11:17:18	OS25	65.37
12/8/2015	11:17:28	OS25	65.87
12/8/2015	11:17:38	OS25	67.12

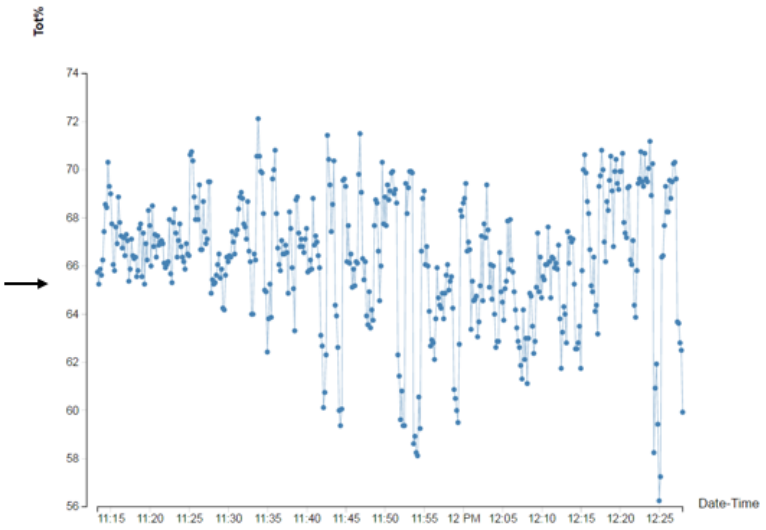

Figure 3.1: Data Mapping of S991DATA to line plot with attributes Date-time and Tot% (Average utilization of all Processors)

There are various versions of line charts [AMST11] like multiline plot, area plot etc., A multiline plot is a grouped variant of basic line chart whereas area plot is a line chart with space between two lines filled with a color. In this project, we will discuss about multiline plot.

3.1.2 Multiline Plot

Most uni variate techniques can be extended to multivariate data. Multiline plots are drawn on common set of axes with difference in line color, style or other graphical attributes. The features of multiline plot are presented in Table 3.2.[AMST11].

An example of single level nesting of SOUT723 data attributes is as shown in the Figure 3.2. In the data table, there are four different attributes like GoalVal, GP, H, Workload. We performed a group by operation on workload attribute for the whole data. The different categories of workload are mapped to the other attributes. Although multiple

levels of nesting is possible for high dimensional datasets, we stick to single level nesting. [AMM+08]

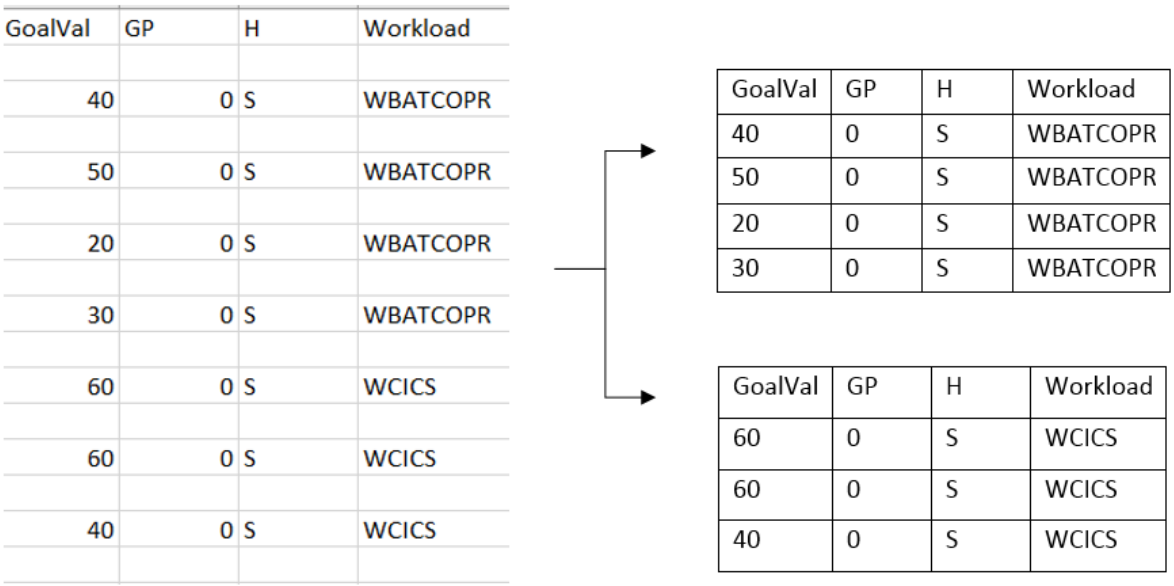

Figure 3.2: Single level nesting of data table based on workload attribute to two key value pairs

Features	Description
Horizontal Axis	Time Attribute
Vertical Axis	Quantitative or Ordered Attribute
Encoding	Expressing values of nested attributes with key attribute and connecting the data points of the grouped variants with a line
Interactions	Details on demand, Nested attribute change, Key attribute change, Axis and label change
Functionality	Finding patterns, Comparison of trends, Understanding relationships

Table 3.2: Features of Multiline Plot

3.1.3 Scatter Plot

Scatter plots are effective for identifying extreme values or anomalies and useful for showing the relationship between two attributes. Sometimes the data points in a scatter plot form distinct groups for different dimensions as shown in Figure 3.3. These distinct groups which are visually identifiable are called clusters[WB97]. The computational way of forming clusters will be discussed in section 4.2.2 through K-means clustering. The features of scatter plot are mentioned in Table 3.3.

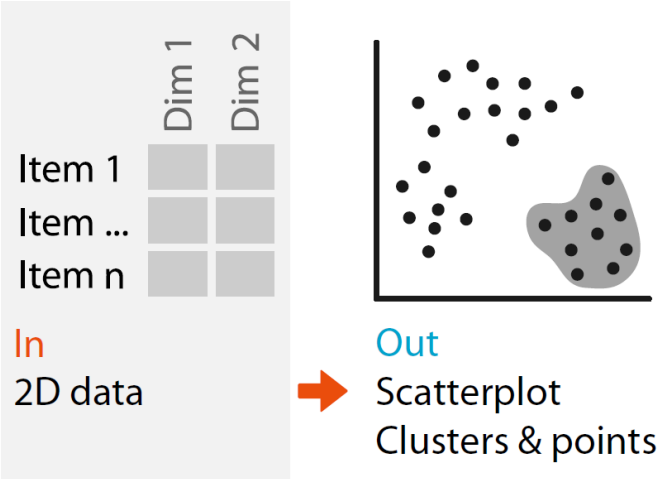

Figure 3.3: Data Mapping to Scatter Plot [MM15]

Features	Description
Horizontal Axis	Quantitative Attribute
Vertical Axis	Quantitative Attribute
Encoding	Expressing values of both the attributes with glyph at corresponding scaled spatial positions.
Interactions	Axis and label change
Functionality	Locate clusters, Correlation identification, Distributions, Finding outliers

Table 3.3: Features of Scatter Plot

Scatterplots are also used for abstract tasks of providing overviews or judging the correlation between two attributes. Various types of correlations can be interpreted through scatterplots. [FWG02]

Positive correlation is aN upward slope (values of both the attributes increase together), Negative correlation is a downward slope (values of one attribute decrease while the other increases). The strength of correlation can be determined by closeness of points in the graph. Correlation is stronger when the points are closer over a diagonal line as shown in Figure 3.4 [War12]

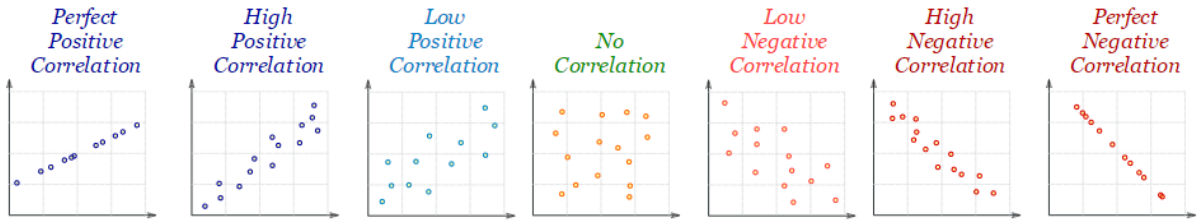

Figure 3.4: Strength of Correlations [mat]

3.1.4 Dot Plot

Dot plot is another variant of scatter plot. Unlike Scatterplot, the horizontal axis is plotted over the time attribute and accompanied with different interaction mechanism like context zooming to filter out the clutter. [MM15]

When user hovers over, the detailed description of the data points are presented. More details of this model will be presented in sections 4.4.2 and in section 5.2

3.1.5 Density Plot

It is based on scatter plot where we use binning technique to find interesting patterns. Binning is a data aggregation technique for visualizing density of data points in large data sets. It is used for grouping a dimension of k values into less than k discrete groups. One way of binning is hexagonal binning which uses hexagon shapes to create bins as shown in the Figure 3.5.[CLM13]

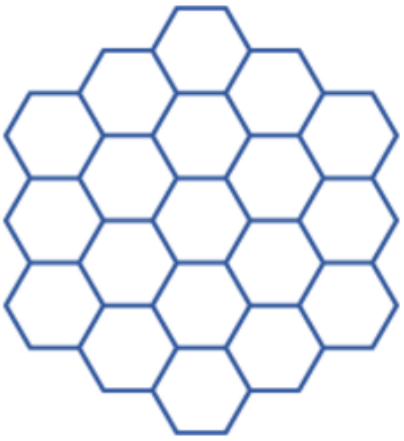

Figure 3.5: Hexbin Structure

We can create a density plot through various shapes like rectangles, squares, triangles. The hexagons are more similar to circles than any other geometric structures. This makes efficient data aggregation around bin center. The distance of vertex points for each of these geometric primitives from their centers is as shown in the Figure 3.6.

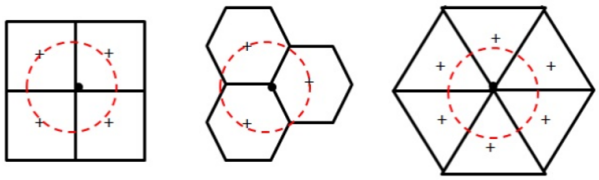

Figure 3.6: Distance of vertex points from centers [Nel]

In this project, we consider binning on Cartesian coordinate system. The system is fairly tiled with polygons and the number of points falling in each bin are counted and stored in an array. The bins with count more than zero are plotted according to the color scales with varying in saturation. [Nel]

Scatter plots are straight forward to visualize data over 2D plane for finding patterns. But, when the dataset is huge, many of these points may overlap which makes it difficult to see patterns or clusters.

Figure **??** is the mapping of the sample dataset where the scatter plot is sparsely spread and Figure **??** is the plot of hexagonal binning for the same dataset. We can observe that density plot clearly shows the patterns which are otherwise hidden in scatterplot. The features of the density plot are mentioned in Table 3.4.

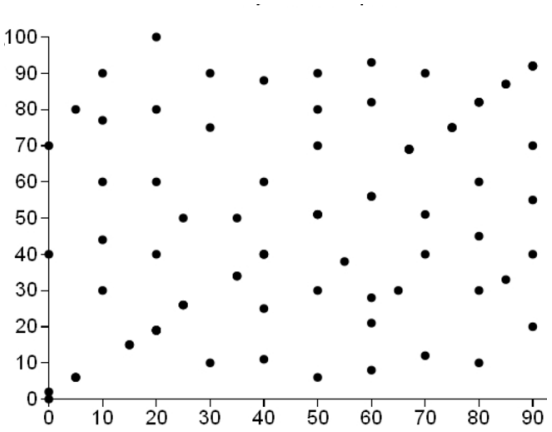

Figure 3.7: An Example of Scatter Plot[Nel]

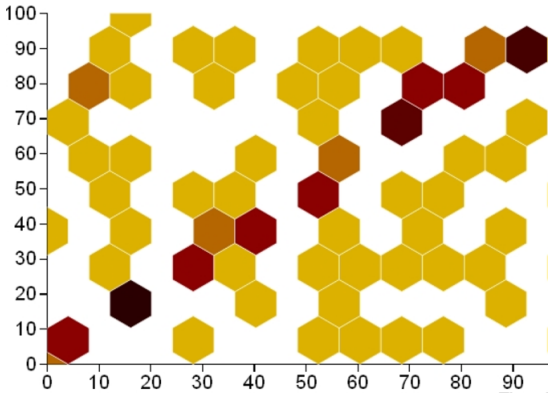

Figure 3.8: An Example of Density Plot[Nel]

Features	Description
Horizontal Axis	Quantitative Attribute
Vertical Axis	Quantitative Attribute
Encoding	Expressing values of both the attributes with hexagonal bins over Cartesian coordinate system
Interactions	Details on demand for count values over each bin, Axis and label change
Functionality	Hidden patterns identification, Data Aggregation

Table 3.4: Features of Density Plot

3.1.6 Correlation Heatmap

Heatmap is created by converting data to a 2D space. For this technique, all the values are mapped to the normalized color space. [BBW16]

For color to be effective, it has to be used sparingly and with color blind people in mind. Too many color variations prevents important data to standout. At the same time, there has to be sufficient contrast to get users attention.

In Figure 3.9, the values from data table are mapped to the heatmap to decrease the mental processing. Here, we used color saturation to make user easily target the points of interest. The range scale of low to high specifies the lighter the saturation of blue

color, lesser the percentage and higher the saturation of color, higher the percentage.The features of correlation heatmap are provided in Table 3.5.[WGK10]

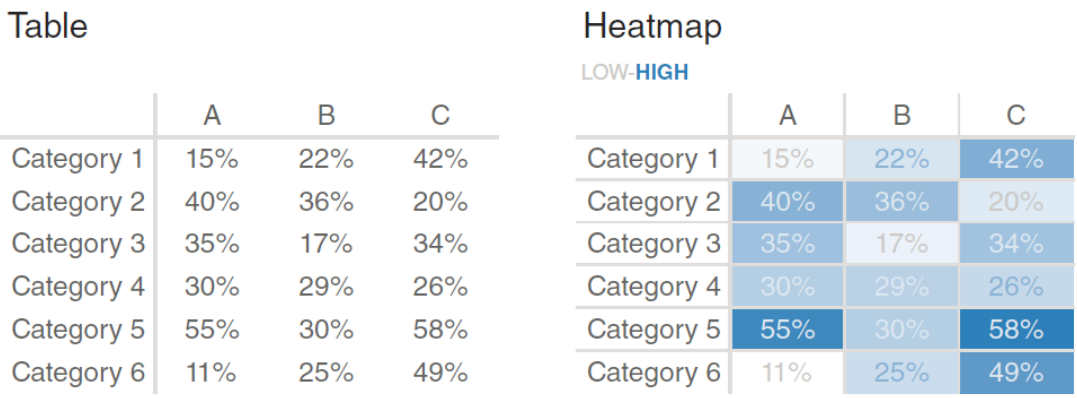

Figure 3.9: Data Mapping to Heatmap [Kna15]

A correlation heatmap provides a compact summary of correlations between attributes in a 2D matrix alignment for a given dataset[ESBB98]. There are various ways of using heatmaps depending on the data at hand. In this project, the correlation values of each attribute with respect to other attributes are calculated prior and these values are mapped to the heatmap. The values with high correlation gets the dense color whereas the weakly correlated value gets the lighter color. This will be discussed in detail in section 4.2.1.

Features	Description
Horizontal Axis	Quantitative Attributes
Vertical Axis	Quantitative Attributes
Encoding	Expressing correlation values of attributes in the dataset with single hue progression
Interactions	Details on demand for correlation values
Functionality	Summarized information

Table 3.5: Features of Correlation Heatmap

3.1.7 Scatter Plot Matrix

A Heatmap shows summarized information whereas scatterplot matrix shows more detailed relationships between attributes. A SPLOM is a matrix with each cell representing a scatter plot. Figure 3.10 shows all the pairwise combinations of four different attributes. [WB97]

The traditional SPLOM requires a lot of display space where each attribute pair is plotted twice once on each side of the diagonal. A more followed practise is to remove the redundant data which is increasing cognitive load. This leads to scatter plot triangle where only the lower triangle or upper triangle is shown rather than redundant full square. The matrix can be reordered according to the user selections.[MM15] A scatterplot triangle is as shown in Figure 3.11

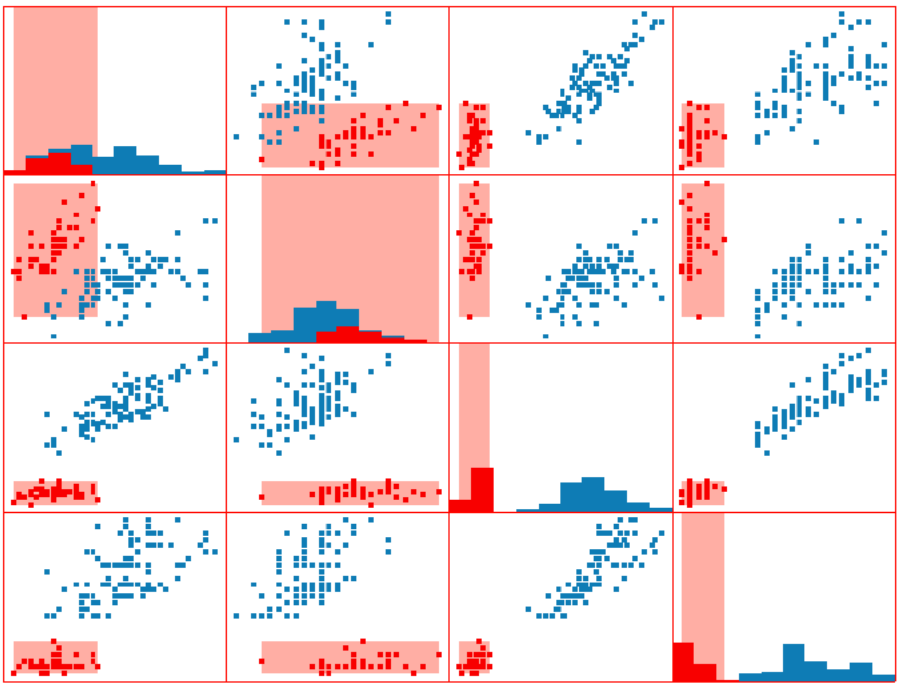

Figure 3.10: An Example of Scatter Plot Matrix [WGK10]

A popular interaction technique is to link the selected data in one view to the corresponding data in the views as shown in Figure 3.11 with red highlighted boxes in all the views. This linked selection highlights a feature in multiple views and reveal interesting features. [BBH+17]

When the data selected is to be interactively changed upon user interest, this is accomplished with brush operator [WB97]. Brushing was first introduced by Cleveland

as one of the manipulation techniques[BC87]. With brushing and linking, user can continuously change the selection by dragging over entire scatterplot and understand the highlighted changes in other views. The features of SPLOM are mentioned in Table 3.6.

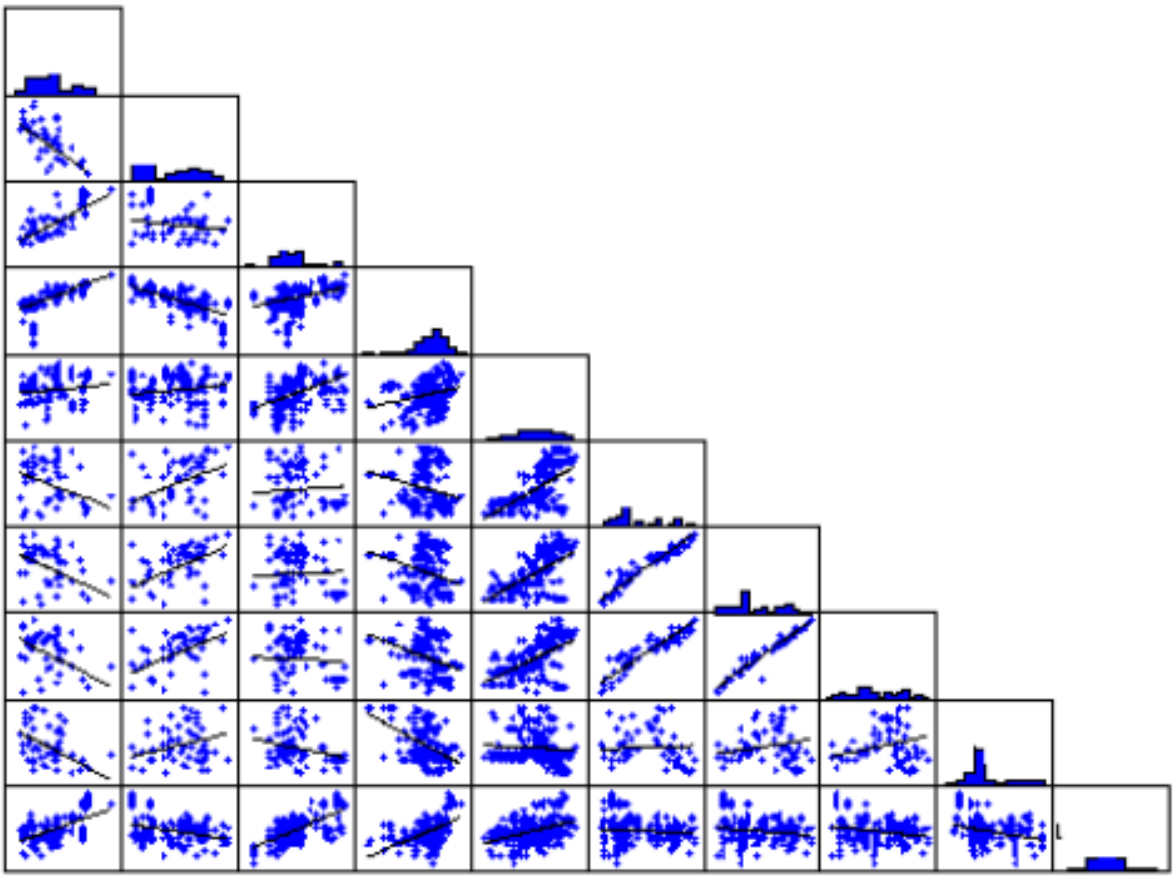

Figure 3.11: An Example of Scatter Plot Triangle [Koc]

A connected SPLOM [BBH+17] is another variant of SPLOM where each data point in a scatterplot is connected with a line to understand patterns between attributes that are changing with time. The implementation details of the same can be seen in section 5.2.[BBW16]

Currently, with SPLOM, the number of scatter plots increase in quadratic manner with the number of attributes. Due to space constraints, the scalability of SPLOM is limited to few attributes.

Features	Description
Horizontal Axis	Quantitative Attributes
Vertical Axis	Quantitative Attributes
Encoding	Scatterplots in 2D matrix alignment
Interactions	Highlighting, Linking and brushing, Matrix order change
Functionality	Finding correlations, Understanding patterns between various attributes

Table 3.6: Features of Scatterplot Matrix

3.1.8 Parallel Coordinates

Unlike SPLOM, parallel coordinates visualization model complexity is directly proportional to the number of attributes. The parallel coordinates is widely used for plotting multidimensional data. Its idea is to map the geometric structures from Cartesian space to parallel coordinate space. As shown in the Figure 3.12, the point line duality principle states that a set of points located on line in data domain are represented by set of lines in parallel coordinates.[GHWG14]

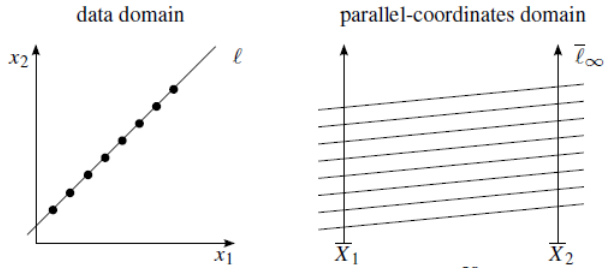

Figure 3.12: Point-Line Duality Principle [HW13]

In parallel coordinates, each vertical axis represents an attribute and every new polyline represents one observation [WB97]. Polyline intersects each vertical axis at point corresponding to the attributes value. For N dimensional dataset, we plot N vertical axes. In Figure 3.13, the dataset has five attributes and the corresponding parallel coordinate plot have five vertical axes with each observation from table being mapped to polyline intersecting the axes. [HW13]

MPG	Cylinders	Horsepower	Weight	Year
15	8	170	3,563	70
14	8	160	3,609	70
15	8	150	3,761	70
14	8	225	3,086	70
24	4	95	2,372	70
22	6	95	2,833	70
18	6	97	2,774	70
21	6	85	2,587	70
27	4	88	2,130	70
26	4	46	1,835	70
25	4	87	2,672	70
24	4	90	2,430	70
25	4	95	2,375	70

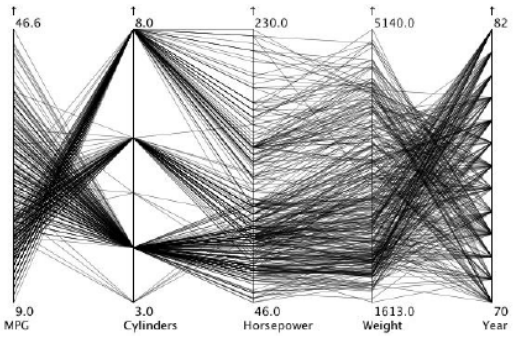

Figure 3.13: Data Mapping to Parallel Coordinates Plot [Koc]

Figure 3.14 shows correlations in parallel coordinates in comparison with correlations in scatter plots. Parallel lines shows perfect positive correlation. Lines crossings over each other at single point in between two axes represents negative correlation. Mix of crossings represents uncorrelated data.

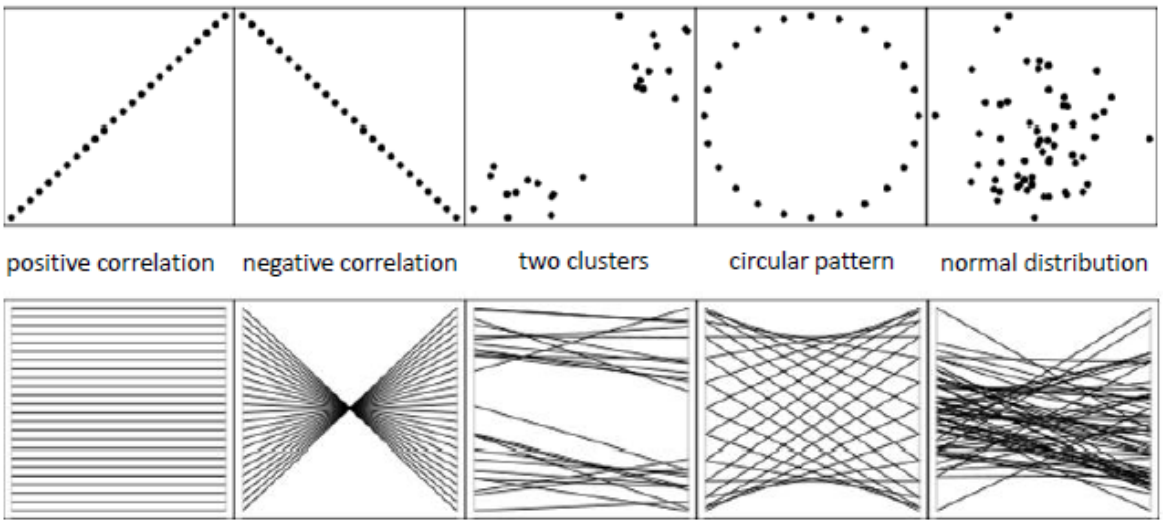

Figure 3.14: Correlations in Parallel Coordinates [Koc]

In general, different models [HSW12] of parallel coordinate plots are composed of different layers. In section 5.2, we see the implementation details of superimposing axes layer over the data layer (polylines). Additive blending and opacity helps parallel coordinate plots to distinguish densely populated observations and outliers.[HB]

Identifying clusters in parallel coordinates plot can be done easily by giving different color code to each cluster as shown in Figure 3.15. We will see about clustering in detail in section 4.2.2 and 5.2

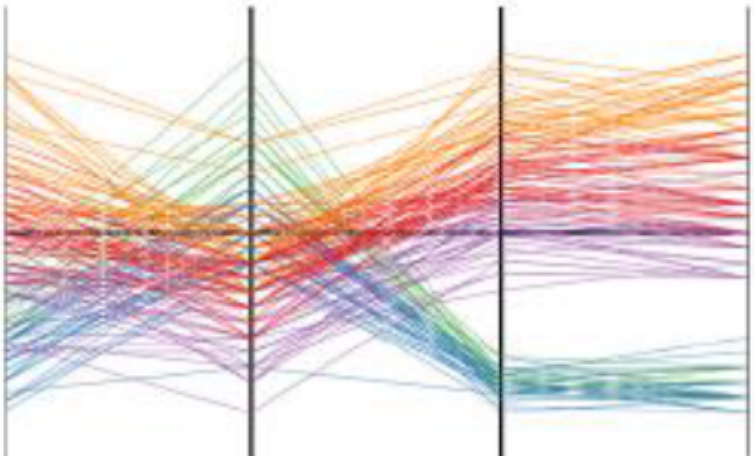

Figure 3.15: Clustered Parallel Coordinates Plot [MM15]

The interactions in parallel coordinates plot are the key control for visual exploration. It allows user to control each stage of the information visualization pipeline depicted in section 2.1.1. There are many interactions possible. In section 5.2, we see in detail about brushing, filtering, axes translation and axes inversion.

Brushing allows user to select data points for highlighting, manipulations. Brushing a point on parallel coordinate axis is equivalent to selection of line in parallel coordinates plot. Figure 3.16 shows sample of axis brushing. [RSM+16]

The axes position in parallel coordinates plot has high impact over patterns emerging from the plot. Axis translation is often implemented with a drag and drop operation as interaction from user. Axes flipping reverses the scaling order and sometimes helpful in understanding the cluttered observations.[HW13]

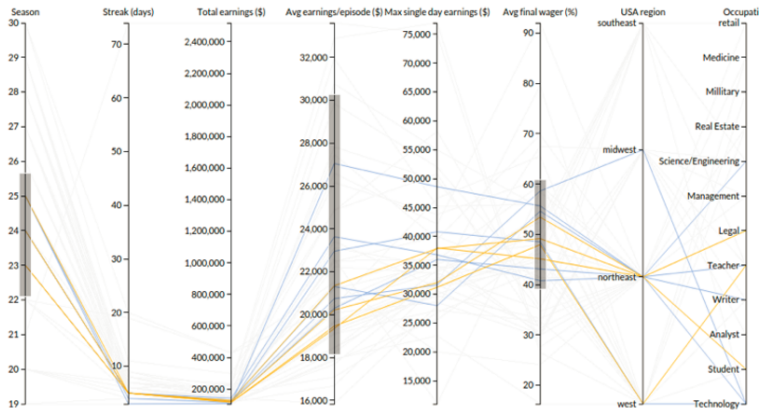

Figure 3.16: Brushing in Parallel Coordinates [Ber]

There are various other techniques that can be employed on parallel coordinates like clustering, data aggregation measures: mean, median, variance etc. We see about them in detail in section 4.2.3. The features of parallel coordinates plot are mentioned in Table 3.7.

Features	Description
Axis	Quantitative or Ordinal Attributes
Encoding	Mapping data points to parallel coordinates space
Interactions	Brushing and linking, Dimension filtering, Axis translation, Axis scaling
Functionality	Finding Correlations, Outliers, Clusters, Scale to many number of dimensions

Table 3.7: Features of Parallel Coordinates Plot

4 Design

4.1 Data Transformation

At first, CSV (comma separated value) is converted to JSON (JavaScript Object Notation) format as shown in the Listing 4.1. JSON is based on two data structures. A collection of name/value pairs as shown in the Figure 4.1 or an ordered list of values as shown in the Figure 4.2.

```
1
2  function Loaddata(file)
3  {
4     var readfile = new FileReader();
5      $( document ).ready(function()
6      {
7          Papa.parse(file,
8                  {
9                      skipEmptyLines: true,
10                     dynamicTyping: true,
11                     complete: function (results)
12                     {
13                         var datum=results.data;
14                         Data_Preprocessing(datum);
15                     }
16                 });
17     });
18
19     readfile.readAsText(file);
20     var filename = file.name;
21     $(".panel-heading").append(filename);
22
23 }
```

Listing 4.1: Parsing and preprocessing the data

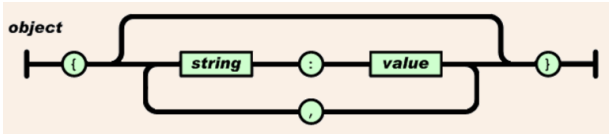

Figure 4.1: Structure of JSON Object [jso]

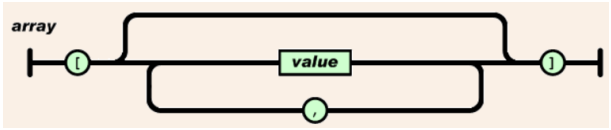

Figure 4.2: Structure of JSON Array [jso]

Processing the dataset in general follow four basic principles:

1. Each attribute has to be in separate column.
2. Each different observation has to be in different row.
3. There should be only one attribute with same name.
4. Understanding quirks and potential errors of dataset. [cou]

There are many data compression algorithms available. These algorithms are chosen depending on the data at hand. In this project, we use run length encoding compression technique.

4.1.1 Run Length Encoding

Run Length Encoding (RLE) is a loss less compression which saves the storage space. It suits for any kind of data sequences.

RLE reduces the data sequences of repeating characters. This repeating character which is called run is stored in two bytes. The first byte gives the information of number of characters/run count and second byte tells about value of character/run value.

For instance, the sequence "hhhhhhhhhh" is interpreted by RLE as 10h where 10 is the run count and h is the run value.

4.2 Analytical Support

4.2.1 Data Correlation

We have seen how correlation can be identified through visual means in section 3.1.3 of scatter plots. A precalcualted correlation factor helps user in understanding his/her visual analysis better.

A correlation factor of 1 represents perfect positive correlation, 0 represents no correlation or the attributes does not seem to be linked at all and a factor of -1 is a perfect negative correlation. [MM15]

In equation 4.1, R is the Pearson correlation coefficient, n is the total number of values, X is the first variable and Y is the second variable. Calculating correlations between these variables is depicted in the Figure 4.3. A first, the mean and variance of these variables are calculated and later these values are substituted in the equation 4.1.

$$R = \frac{n(\sum XY) - (\sum X)(\sum Y)}{\sqrt{[n\sum X^2 - (\sum X)^2][n\sum Y^2 - (\sum Y)^2]}} \tag{4.1}$$

	1		2		3		4	
	X	Y	X	Y	X	Y	X	Y
	10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
	8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
	13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
	9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
	11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
	14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
	6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
	4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
	12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
	7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
	5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89
Mean	9.0	7.5	9.0	7.5	9.0	7.5	9.0	7.5
Variance	10.0	3.75	10.0	3.75	10.0	3.75	10.0	3.75
Correlation	0.816		0.816		0.816		0.816	

Figure 4.3: Calculating Correlation from Data Variables [MM15]

4.2.2 Data Clustering

There are various ways to cluster the dataset. In this project, we see about two basic ways of clustering. One way is to cluster according to the categorical values of attribute. This is a simple technique of assigning different color to each category of attribute and enable interaction on top of the visual model. User can select the categorical attributes user interested to see clusters and perform interactions on the model according to the exploration objectives. [Hua13]

Another way of clustering is to apply a clustering algorithm before visualizing the model. There are various clustering algorithms like k-means[KMN+], agglomerative, birch etc[sci]. We shall see about k-means in detail. K-means algorithm clusters the data into x groups where x is predefined.

The K-means algorithm is basically divided into following five steps: [wika]
1. Cluster the dataset into x groups.
2. Choose x sample points as random cluster centers.
3. Assign objects to the nearest cluster according to euclidean distance.
4. Set the centroid position as new cluster center.
5. Algorithm terminates when no points changed the cluster between two iterations.

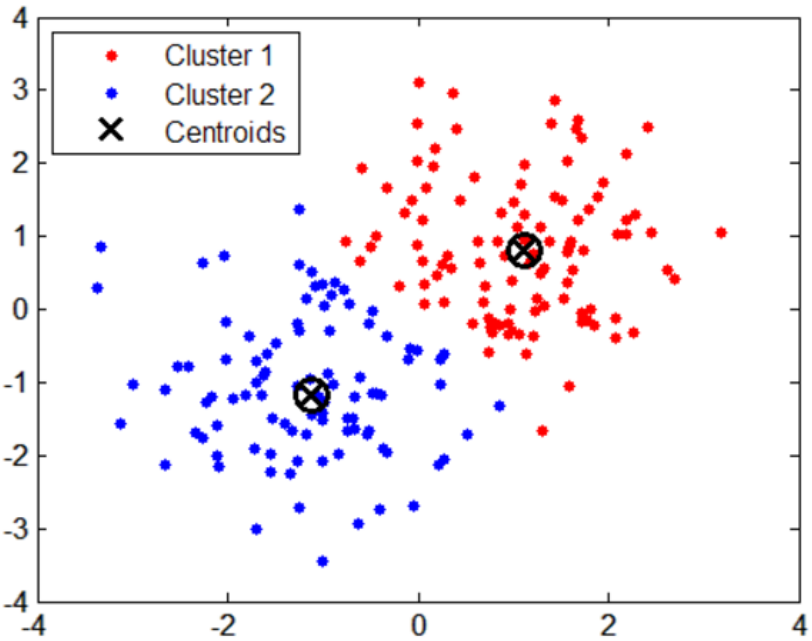

Figure 4.4: Standard Clustering using K-means Algorithm [dem]

The algorithm is depicted in more comprehensible way in Figure 4.4 The dataset is first grouped into two clusters (cluster1 and cluster2) and objects are assigned to these clusters. We shall see about the implementation details of k-means algorithm over parallel coordinates in section 5.2.

4.2.3 Data Aggregation

There are various ways to reduce clutter in high dimensional data. They are mainly categorized by two ways. One is through data driven approach and other by visual approach. We shall see about visual approach in section 4.4.2. [HW13]

Data driven approach refers to algorithms that operate on data before mapping them to the visualization model[MM15].

There are various ways of aggregating the data and render the aggregated items. Typical statistical measures like mean, median or variance are performed over the observations and the aggregated items are displayed to the user. In section 5.2, We see about the implementation details of the data aggregation with mean over the K-means clustered samples in parallel coordinates visual model. [HW13]

4.3 Data Rendering

For visualization on web, there are various rendering techniques. Each of their pros and cons are discussed in later sections.

4.3.1 SVG

SVG is a scalable vector graphics which supports interactivity over web. For SVG, there will be no loss of quality irrespective of zooming or reshaping the graphics as shown in Figure 4.5.[Wik]

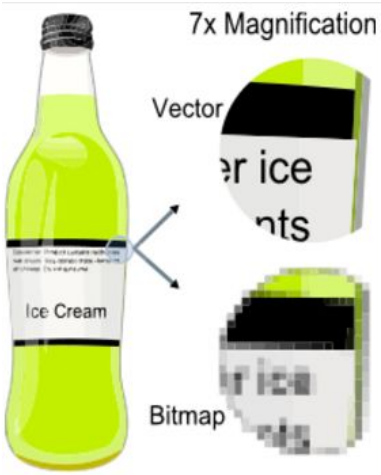

Figure 4.5: Comparison of SVG and Raster Graphics [Wik]

As SVG is vector based, every shape drawn is attached to DOM and if attributes of SVG object are changed, browser re-renders the scene automatically. A sample snippet of rendering circle with SVG is shown in Listing 5.3. [tut]

```
<svg width="100" height="100">
    <circle cx="100" cy="100" r="40" stroke="black" stroke-width="5" fill="pink" />
</svg>
```

Listing 4.2: An example of rendering circle with SVG

However, for huge datasets, SVG will be slow to render all the items and sometimes browser crashes due to heaviness of DOM elements. Compared to SVG, the performance is better with canvas element.

4.3.2 HTML5 Canvas

HTML5 <canvas> element enables us to render graphics with the help of JavaScript. It basically takes three attributes like id, width and height of canvas. Unlike SVG, if attributes position is changed, whole scene need to be redrawn.[tut]

a) Canvas 2D

For 2D canvas, we call the DOM method getContext() within the context 2D as shown in Listing 5.4.

```
1
2 <script>
3     var canvas = document.getElementById('canvas');
4     var context = canvas.getContext('2d');
5     context.fillText('Visualization', 70, 70);
6 </script>
```

Listing 4.3: A sample snippet of canvas 2D context

b) WebGL

Compared to HTML5 2D canvas, WebGL performs much better as it works directly with the graphics card using OpenGL ES. WebGL has basically two shaders as shown in the Figure 4.6: vertex shader and fragment shader. Vertex shader provides the vertex positions and fragment shader provides the color. [Øye15]

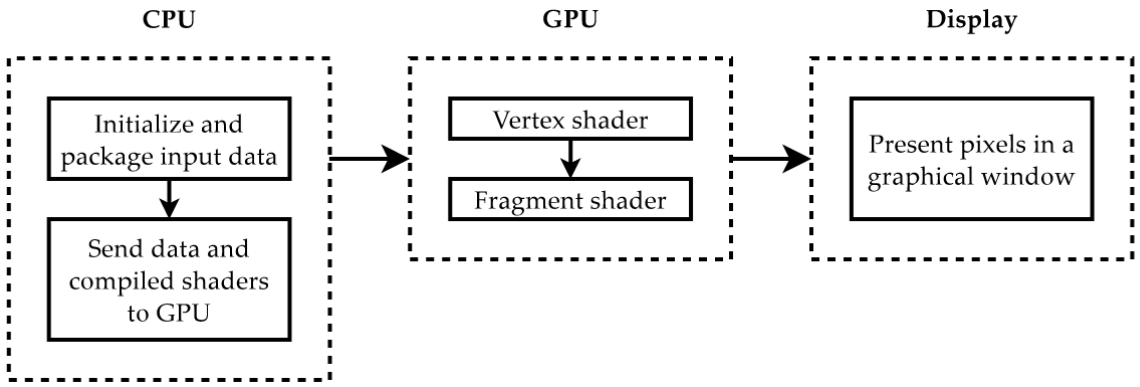

Figure 4.6: WebGL Rendering Pipeline[Øye15]

Rendering image using WebGL follow below basic steps: [tut]

Step 1: Select the WebGL Context in the canvas
Step 2: Define and store the geometry in vertex buffer objects
Step 3: Create a vertex and fragment shader programs
Step 4: Compile the shader programs
Step 5: Create shader program to store the combined shaders
Step 6: Link the shader programs to vertex buffer objects
Step 7: Draw the line object

These steps are implemented with simple example of rendering line objects in Listing 4.4.

```
<script>
    var gl = canvas.getContext('experimental-webgl',{
        alpha             : true,
        antialias         : true,
        depth             : true,
        stencil           : false,
        premultipliedAlpha: false
    });

    var vertices = [0.0, 1.0, 0.1, 0.5, 0.2, 0.3, 0.3, 0.2, 0.4, 0.3, 0.5, 0.9,
        0.6, 0.7, 0.7, 0.2, 0.8, 0.4, 0.9, 0.3];

    var vertex_buffer = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, vertex_buffer);
    gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(vertices), gl.STATIC_DRAW);
    gl.bindBuffer(gl.ARRAY_BUFFER, null);

    var vertCode =
        'attribute vec2 coordinates;' +
        'void main(void) {' + ' gl_Position = vec4(coordinates,0.0, 1.0);' + '}';

    var vertShader = gl.createShader(gl.VERTEX_SHADER);
    gl.shaderSource(vertShader, vertCode);
    gl.compileShader(vertShader);

    var fragCode = 'void main(void) {' + 'gl_FragColor = vec4(0.0, 0.0, 0.0, 0.1);' + '}';

    var fragShader = gl.createShader(gl.FRAGMENT_SHADER);
    gl.shaderSource(fragShader, fragCode);
    gl.compileShader(fragShader);

    var shaderProgram = gl.createProgram();
    gl.attachShader(shaderProgram, vertShader);
    gl.attachShader(shaderProgram, fragShader);

    gl.linkProgram(shaderProgram);
    gl.useProgram(shaderProgram);

    gl.bindBuffer(gl.ARRAY_BUFFER, vertex_buffer);
    var coord = gl.getAttribLocation(shaderProgram, "coordinates");
    gl.vertexAttribPointer(coord, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(coord);
    gl.clearColor(0.5, 0.5, 0.5, 0.9);
    gl.viewport(0,0,canvas.width,canvas.height);
    gl.drawArrays(gl.LINES, 0, 10);
</script>
```

Listing 4.4: An example of rendering line objects within WebGL context [tut]

4.4 Interaction

Interaction is the key factor for visual analysis. It enhances the user perception for dataset exploration. It enables the change of parameters interactively upon user selection and returns the immediate feedback from the system [HW13]. There are various interaction techniques possible. We see about them in detail in later sections.

4.4.1 Brushing and Linking

Brushing in visualization is the means of selecting subset of data items from the visual display.

The concept of linking and brushing is an important factor for interactive visual exploration. Brushing makes user to select subsets of data interactively and linking connects the selection in other views. With the brushed result, user can perform multiple operations like visual highlighting, defining data for another visual model, extracting the selected user samples from JSON to CSV for further processing and so on. [BC87]

Brushing techniques can be broadly classified as three categories: screen brushes, data brushes and structure brushes.[RSM+16]

The structure of screen shaped brushing is limited to two dimensions. This kind of brush is implemented in scatter plot matrix. The data space techniques allows brushes to scale for N-Dimensions. For High dimensional data, using a single brush is not sufficient. Users need to perform multiple brushes on different attributes or in different views to support a visual query mechanism. [RSM+16]

4.4.2 Zooming

As the amount of data to be presented is limited by resolution of display, zooming is important to overcome this limitation. There are various forms of zooming possible like geometric zoom, fisheye zoom and semantic zoom. [wikf]

The standard geometric zooming allows user to increase or decrease the image with scale of magnification. This view depends on physical properties of what is being viewed. In this technique, objects appearance is fixed and changing viewpoint changes size of objects at image space. Geometric zooming focuses on specific area and information outside this area is discarded. Fisheye zoom is also similar to geometric zoom except information outside the area is not lost from view but distorted.

Unlike geometric zoom, semantic zooming changes the context in which the information is presented. The visual appearance of object changes drastically at various scales. An example of this technique is implemented in dot plot which will be presented in section 5.2 where the user brushes the region he/she wants to zoom. [wikf]

In semantic zoom normal view, the time scale shows the hours of the day. When user zoom in, the scale alters at the point where he/she brushed from hours to minutes. When the user zoom out, view comes to normal form. In this method, the actual information did not change, only the presentation method changed.

4.4.3 Colormaps

Colormaps are used for visual data filtering. It is a mapping between colors and data values. Colormaps can be categorical or ordered.

Both luminance and saturation colormaps are used for ordered data whereas hue is used for categorical data.[MM15] The difference between them is shown in Figure 4.7

In correlation heatmap, saturation color scales from Colorbrewer [CUb] are used which incorporates effective guidelines into its design to provide safe suggestions.

Figure 4.7: Comparison of Luminance, Saturation and Hue [MM15]

4.4.4 Opacity

Opacity is opposite of transparency. This is mostly used in combination with superimposed layers to make foreground layer distinguishable from background. The opacity of the color can be set with the alpha parameter that range from 0.0 to 1.0 where 0.0 is fully transparent and 1.0 is fully opaque. [MM15]

Choosing low opacity value highlights the recurring observations to standout and outliers to filter out. Opacity interacts closely with luminance, saturation and cannot be used in conjunction with them. However, it can be used with hue encoding.[WGK10]

4.4.5 Tooltip

Mouse hover is the most common action performed by dragging cursor over a data object. The extra information about a geometric primitive like name or type is revealed when mouse cursor passes over it. Usually it is implemented with a delay as shown in Listing 4.1. [BBH+17]

```
on("mouseover", function() {
    tooltip.transition()
        .duration(10)
        .style("opacity", 1);
    tooltip.html(name)
})
on("mouseout", function() { tooltip.transition().style("opacity", 0);});
```

Listing 4.5: A sample snippet for hover over data

4.4.6 Highlight

This mechanism focuses user attention on subsets of data. The selections chosen are indicated by changing their visual appearance in some way. This is usually done with change in color or shape or size. One example is highlighting particular subset and linking its view to other plots.[BBH+17]

Highlighting provides immediate visual feedback to make sure that the results of user operations match with their intentions. In section 5.2 , we see highlighting in scatter plot matrix with a shadow mode as a technique to remove clutter.[IS11]

5 Implementation

5.1 Web Production Tools

5.1.1 Node Package Manager

Node Package Manager (NPM) runs on top of Node.js. Node.js is built on chromes JavaScript run time for easily building fast, scalable network applications[nod]. NPM manages the dependencies required for this project. Once node is installed, we install tools like brunch through NPM from command prompt. [RFP15]

Our platform is mainly classified into folders as shown in Figure 5.1.

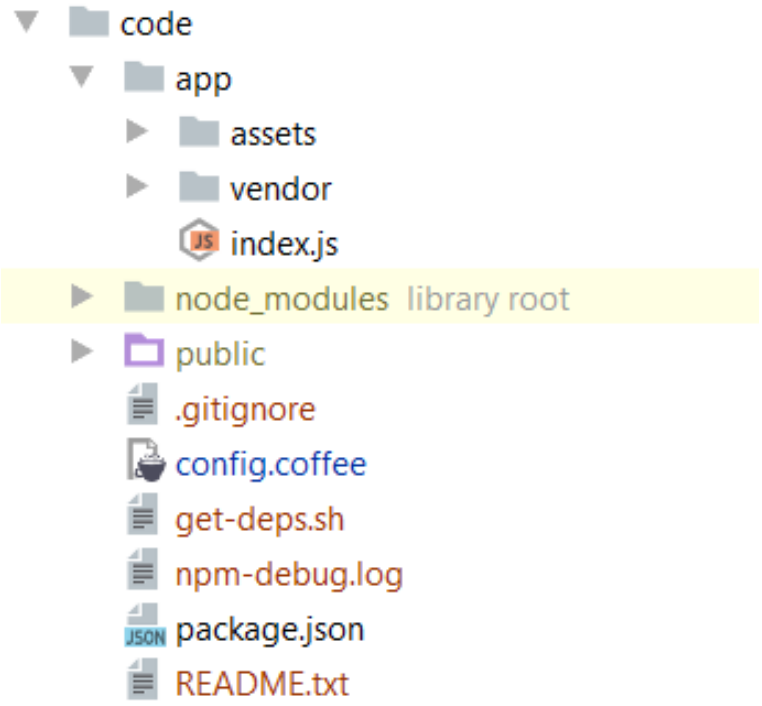

Figure 5.1: Folder Structure of the Project

App Folder: It contains the modules for data connection and essential API's.

Node_modules Folder: It holds various node modules necessary for the functionality of the server.

Public Folder: This contains various JavaScript, HTML, CSS files required for rendering the visual model.

Package.json: This file has list of all modules needed to be installed for the server to run.

5.1.2 Brunch

Brunch is a build tool that automates unit testing and linting required for this project. [RFP15]

```
#
# Brunch configuration file.
#
exports.config =
  paths:
    watched: [
      'app'
    ]

  files:
    javascripts:
      joinTo:
        'js/app.js': /^app(\/|\\)(?!vendor)/
        'js/vendor.js': /vendor(\/|\\)/

      order:
        before: [
          'app/vendor/js/underscore.js'
        ]
    stylesheets:
      joinTo:
          'css/app.css': /^app[\\/](?!vendor)/
          'css/vendor.css': /^app[\\/]vendor/

    templates:
      joinTo: 'js/app.js'

  server:
    port: 9000
```

Listing 5.1: Brunch configuration file

The Listing 5.1 expands the config.coffee brunch configuration file specified in the Figure 5.1. In the configuration file, paths specifies from where to take files and where to put the generated ones, files tells which files should brunch generate and how, server specifies the port location. [bru]

5.1.3 Subversion

Subversion(SVN) is a version control system to keep track of changes in the code. It is used for checking the modifications that were made to the source code and to revert back to the earlier versions in case of errors. Version control systems provides log for software development. [RFP15]

5.2 Visual Analysis of SMF Records

In this section, we shall see about the implementation details of the visual models discussed in section 3.

5.2.1 Dashboard

The user interface is basically divided in to three main parts. The first part is for taking records as input from user. The second part is a panel attached to the left of the dashboard which provides axis selections for various visual models. The third part is the space left for rendering visual models.

From the Figure 5.2, you can observe that user selected the file SOUT723 whose name can be identified at the header of the panel. Below the header, there exists a drop box for visual model selections, axis selections of various visual models and the pre filtered attributes. In this example, 21 out of 71 attributes are pre filtered from the dataset SOUT723 according to the RLE discussed in section 4.1.1. The attributes and their values can be observed from the Figure 5.2. This filtering answers our question 2.7 in section 2.2.4.

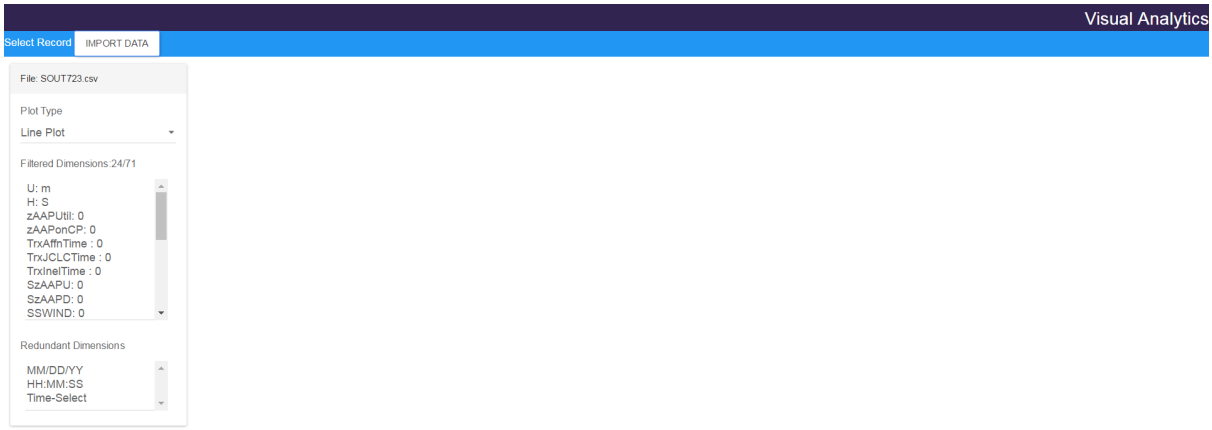

Figure 5.2: An Overview of the Dashboard

5.2.2 WLM Records and Visualization Scenarios

As specified in section 2.2, we have WLM records from subtypes 1-14. In this section, we will see the visual analysis performed on some of them.

5.2.2.1 Line Plot of S991DATA

S991DATA record has samples that are monitored for every 10 second interval. This record has 84 dimensions with 36900 data points. Out of those, 42 dimensions are pre filtered and displayed on the panel with attribute name and its value. The visual analysis will be carried out with the remaining 40 crucial dimensions.[FWG02]

In this line plot, we see how the average percentage of regular CP's (CPU%) is changing with respect to the time. We observed that most of the time percentage is at 100 whereas when time intervals are reaching peak values like 11:55AM, 12:25PM, comparatively less percentage of CP's are consumed as shown in Figure 5.3.

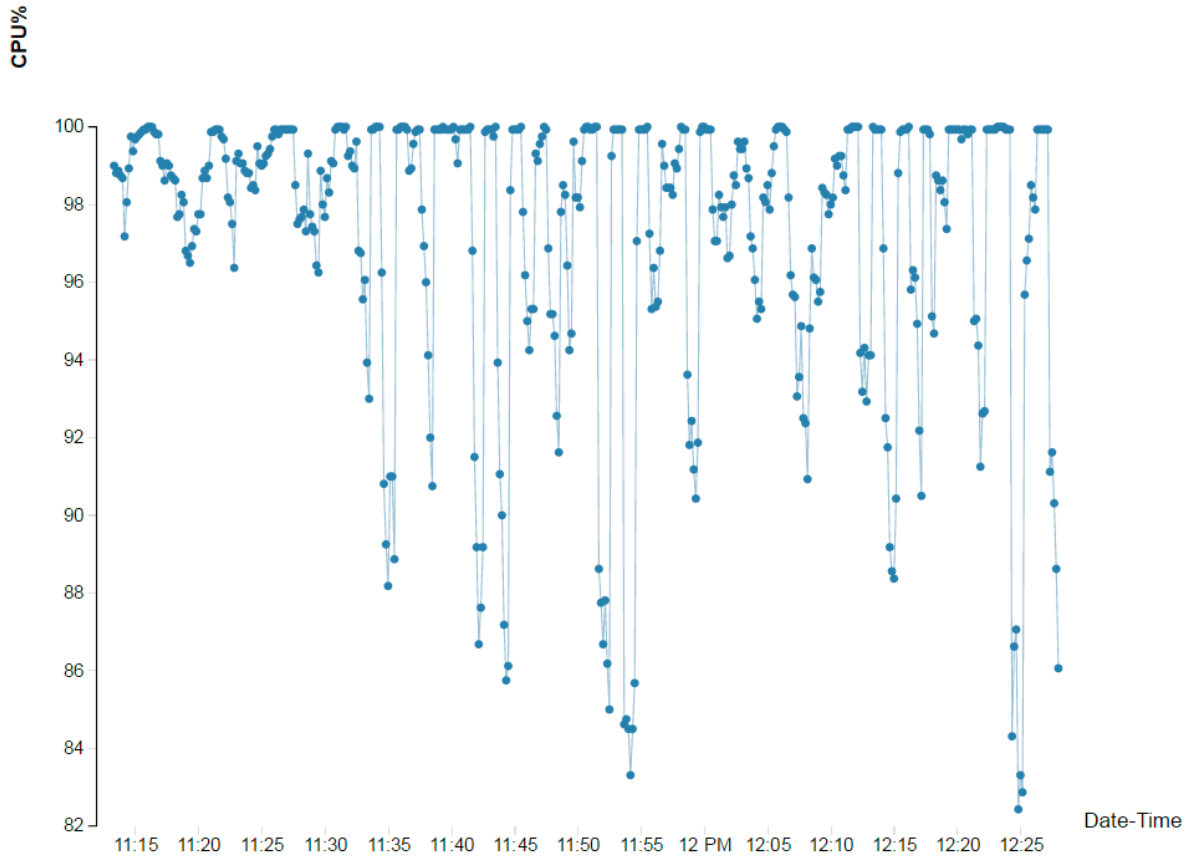

Figure 5.3: Evolution analysis of CPU% in S991DATA with line plot

User can see how other attributes are behaving with time by choosing the desired attribute from the panel y axis selection. The scales automatically get updated each time user selects the attributes. For above selection with CPU(%), scales are ranging from 82 to 100. When user changes the attribute to INFACT from the yaxis drop down menu in the panel, scales got updated and we can observe from the Figure 5.4, there is outlier at 128 that deviates from the normal flow which usually lies between the range 104 and 109.

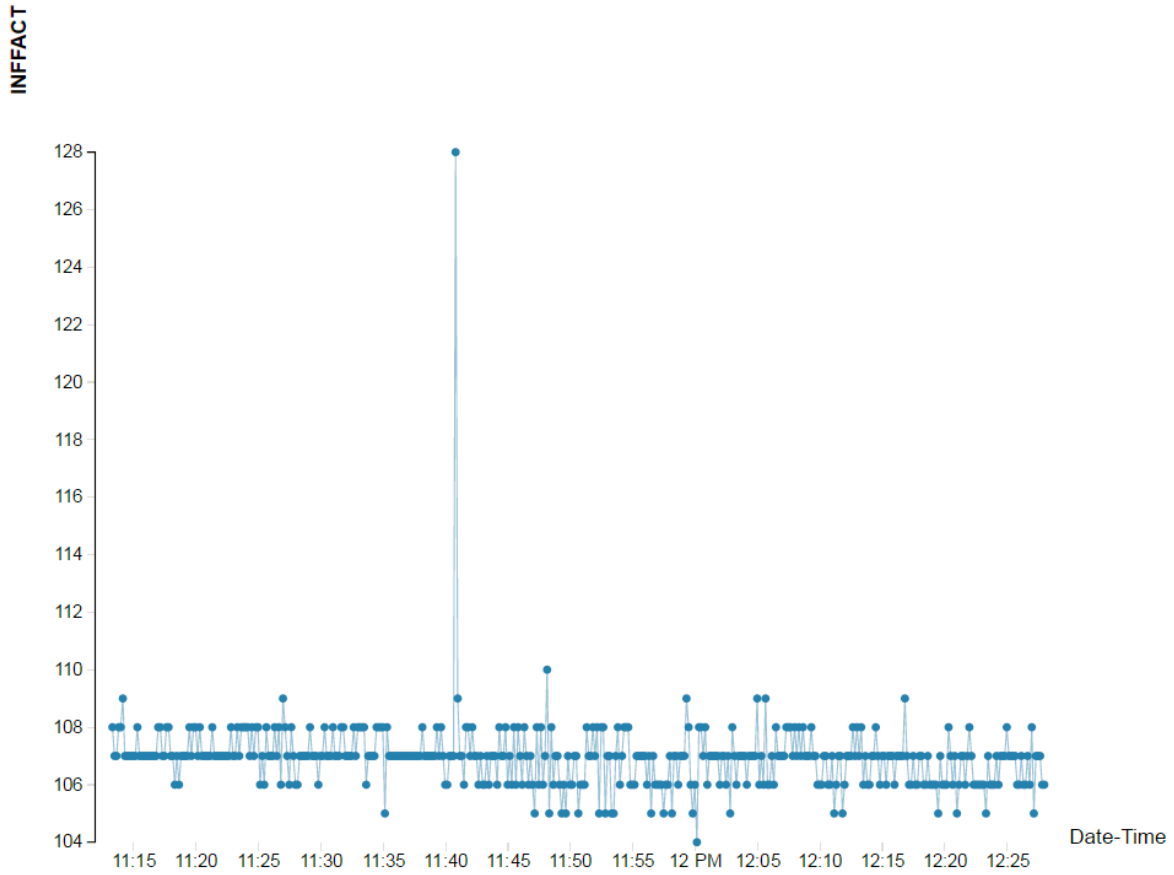

Figure 5.4: Identifying outliers in attribute INFFACT of S991DATA with line plot

After exploring through various attributes of line plot, we answered our question 2.2 which raised in section 2.2.4 during evaluation of records.

Component	Functionality
Horizontal Axis	Time interval for the S991DATA record
Vertical Axis	Dynamically changing attributes upon user selection
Interaction	Updating labels and scales in correspondence to the encoding

Table 5.1: Functionality Description of Line Plot Components

5.2.2.2 Correlation Heatmap of S991DATA

We can observe from Figure 5.5, attributes Tot% and CPU% are in high correlation and all the attributes ZiiCpWai, ZiiCpGra, ZiiCpCEC, ZiiCpLCP, ZiiFreCp are in high correlation with attribute SUpFree. This summarized relation between attributes answers over question 2.4 in section 2.2.4.

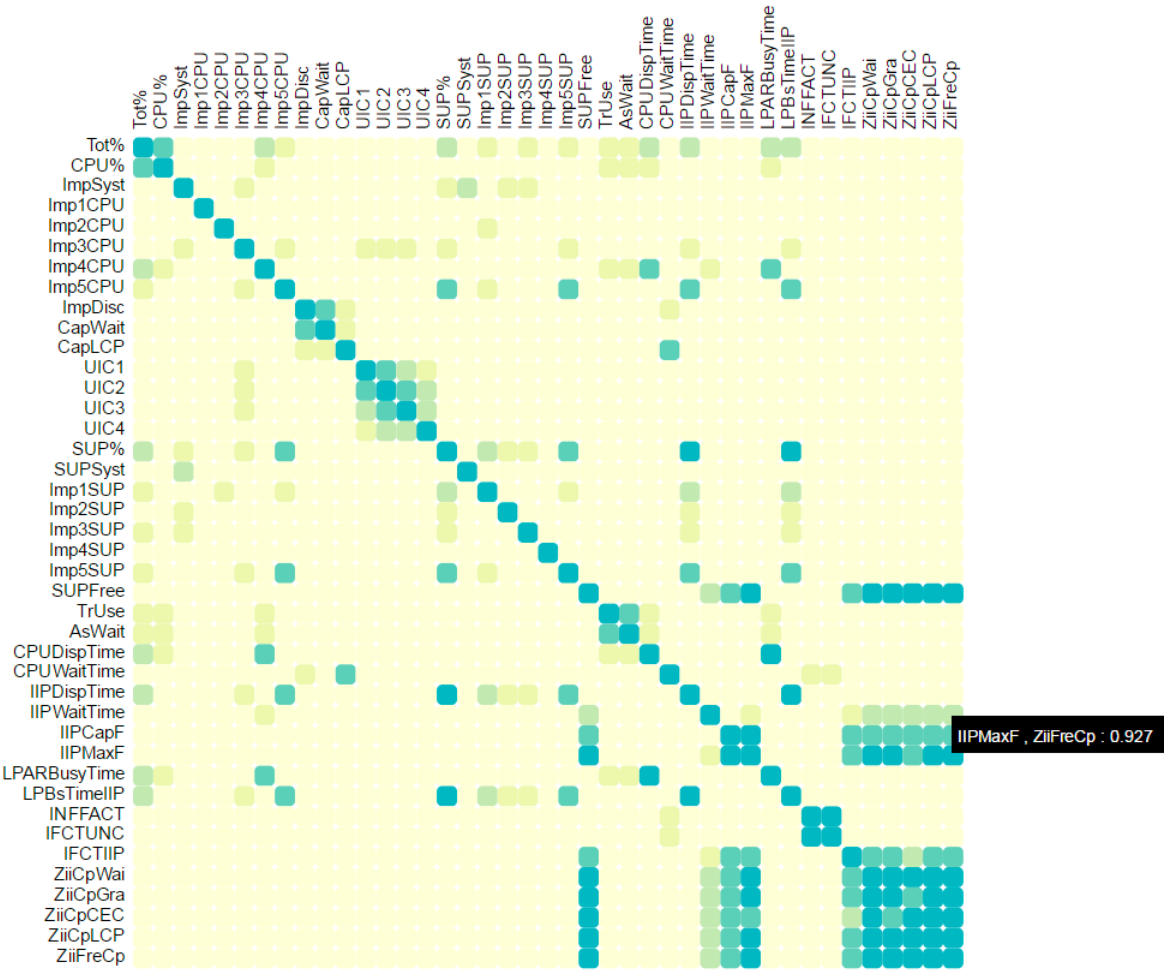

Figure 5.5: Summarized correlations between attributes through correlation heatmap

5.2.2.3 Scatter Plot of S991DATA

Though correlation heatmap helps user to get a summarized view, correlation is not always the causation. Visually exploring through scatter plot and understanding the clusters or patterns within the attributes is necessary. With the quick glance of heatmap,

Component	Functionality
Horizontal and Vertical Axis	All the quantitative attributes of the record
Interactions	Filtering the strongly correlated values from weak correlations through saturated color scales where dark blue shows high correlation of 1.0 and saturation decreases with value moving towards -1.0. Mouse hover the attribute pairs to see their strength of correlations with the pearson correlation value.

Table 5.2: Functionality Description of Correlation Heatmap Components

user exploration becomes much easier on which attributes he has to concentrate further for more visual analysis through scatter plot.

Figure 5.6 shows us the side by side view of scatter plot and the correlation heatmap. The scatter plot between attributes Tot% and CPU% shows us that both are linearly correlated and also points out the outliers or densely distributed regions. This also answers our question 2.9 and 2.10 from section 2.2.4.

Figure 5.6: Visual analysis of correlations between attributes CPU% & Tot% in S991DATA through scatter plot

User can change both the vertical and horizontal axis in scatter plot and see if the relations are matching his/her expectations.

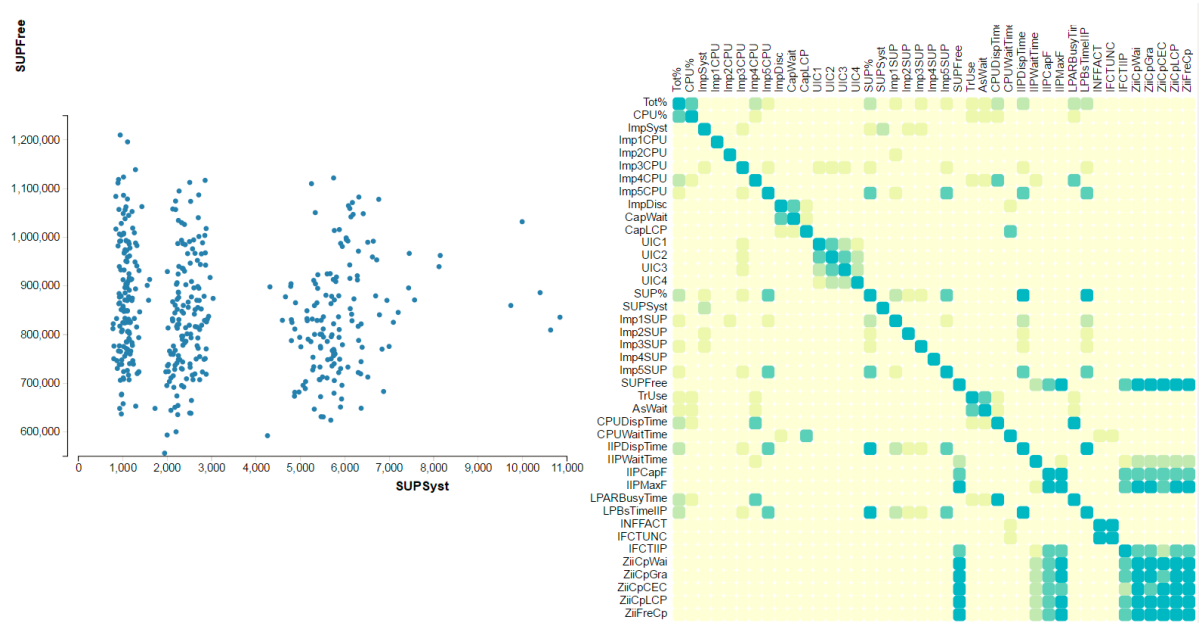

Figure 5.7: Visual identification of clusters between attributes SUPFree & SUPSyst in S991DATA through scatter plot

In Figure 5.7, both the vertical and horizontal attributes are changed to SUPFree and SUPSyst. Though these attributes are negatively correlated from heatmap, you can identify three different clusters when visualized in scatter plot apart from few anomalies.

Component	Functionality
Horizontal Axis	Selection of attributes from x axis of panel
Vertical Axis	Selection of attributes from y axis of panel
Interactions	Change the axis scales, labels of the plot. Optional help of correlation heatmap.

Table 5.3: Functionality Description of Scatter Plot Components

5.2.2.4 Scatter Plot Matrix of S991DATA

Scatter plot matrix as discussed in section 3.1.7 is implemented with scatter plot triangle neglecting the repeating plots. User selects multiple attributes from the panel. In Figure 5.8, we selected four attributes which belong to one category. User can deselect by clicking twice on the same attribute from the multi attribute selection choice in the panel. [FWG02]

UIC buckets from one till four are plotted against one another. The diagonal plots are plotted over the same attribute hence a linear line is seen all the time. In Figure 5.8, we performed two dimensional brush over the scatter plot of UIC1, UIC3 and linked this view to other scatter plots.[BC87]

Moving the brush throughout the plot and understanding how patterns are changing with the selection in other views is helpful for identifying the relationship between specific set of attributes[BC87]. This answers our question 2.8 from section 2.2.4. User can undo the brush operator by clicking elsewhere on the plot. Here, keeping the space and complexity of plot in view, the plots are decreased in size when the attributes are increasing.

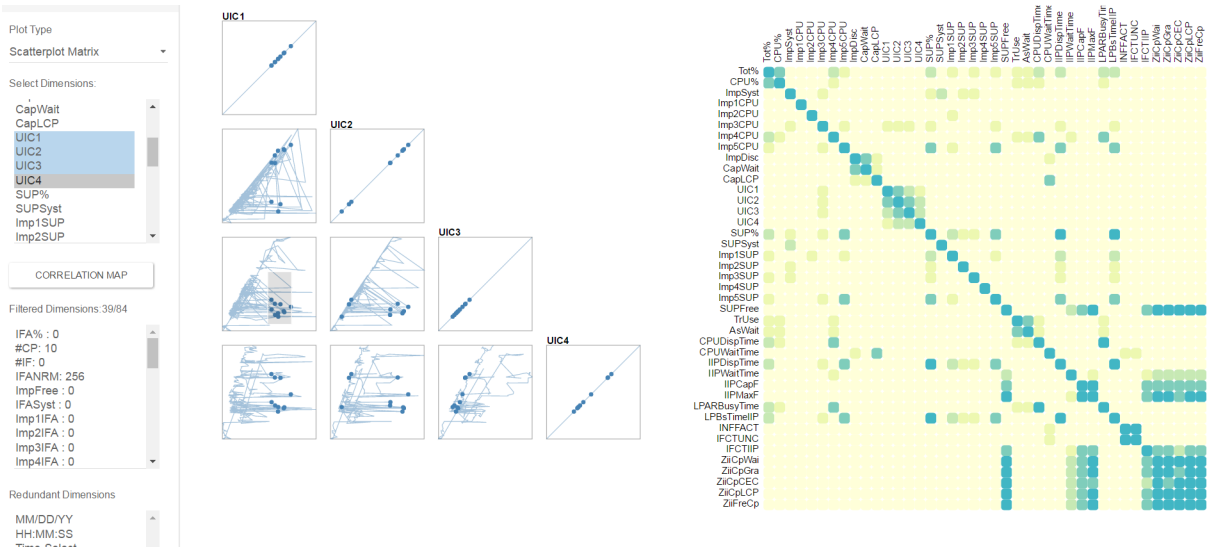

Figure 5.8: Brushing and linking the views of attributes UIC1, UIC2, UIC3, UIC4 in S991DATA through scatter plot matrix

5.2.2.5 Dot Plot of S992DATA

S992DATA record contains 108 dimensions with more than a million data points. Dot plot enables us to remove the clutter in huge datasets through zooming technique we

discussed in section 4.4.2. The Figure 5.9 between time attribute and attribute B0950 shows us the densely distributed regions. We can perform the zoom operation on this plot and get the details on demand through mouse hover. [MM15]

Component	Functionality
Horizontal Axis	Time scale of the record
Vertical Axis	Selection of categorical or quantitative attributes from y axis of panel
Interactions	Zooming the time scale. Details on demand.

Table 5.4: Functionality Description of Dot Plot Components

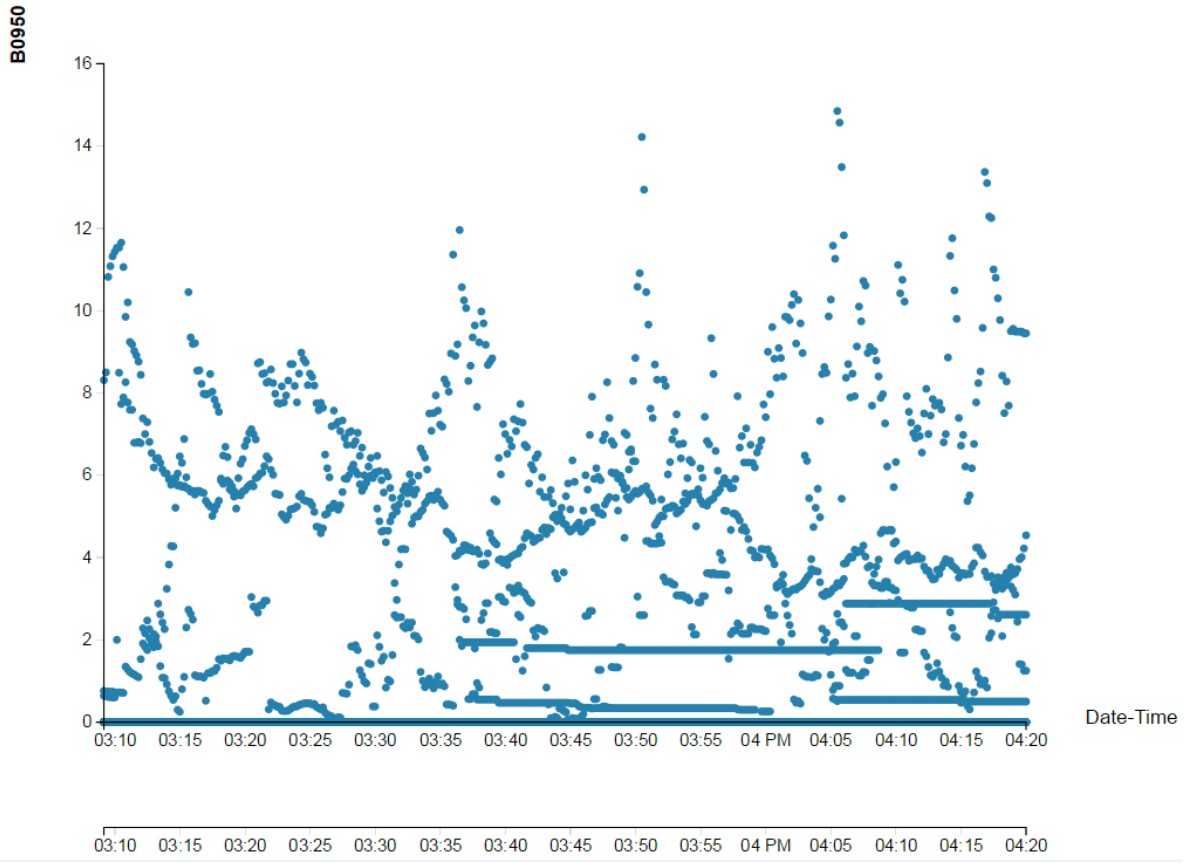

Figure 5.9: Cluttered observations view of attribute B0950 in S992DATA through dot plot

In Figure 5.10, we can observe two scales of x axis. One scale is used to maintain the overall context and other scale belongs to the main plot for focusing the brushed selection from the user. In this plot, user performed selection between time 3.35 to 3.40 for the attribute B0950.

The time scale got expanded from minutes to seconds for zoomed in view of the plot. User can click elsewhere on the axis to undo the selection (zoom-out operation) and get back to the original plot for further exploration.

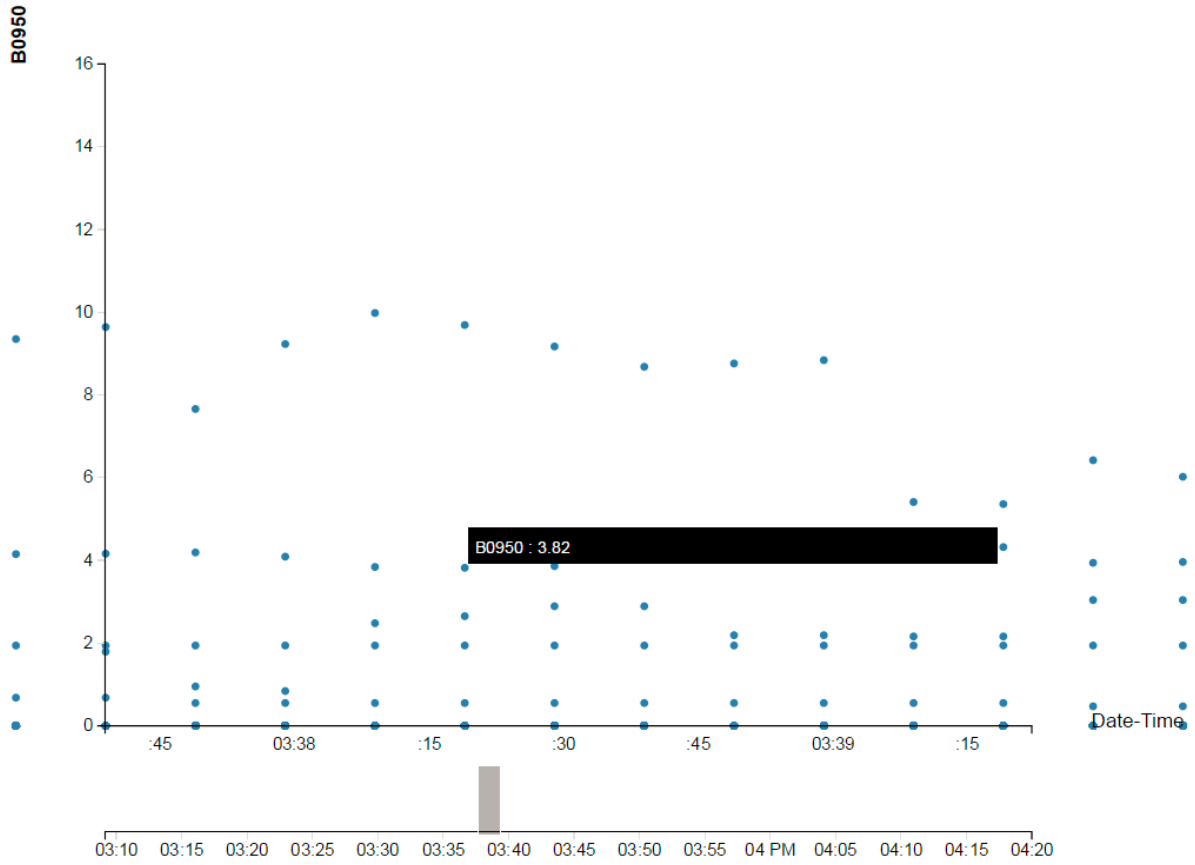

Figure 5.10: Performing the zoom operation over the cluttered observations of attribute B0950 in S992DATA

5.2.2.6 Density Plot of LFLTOSO7

LFLTOSO7 record has observations taken for every 10 minute interval and has 46 dimensions with 767602 data pints. Density plot between attributes Partition LCP's and LPAR Number shows us the patterns that are not so visible through scatter plots.

For instance, in the Figure 5.11, you can observe that hexbins are colored according to their recurring interval in the dataset. Mouse hover on violet colored hexbin shows us that particular pattern is repeated for 432 times in the dataset. Similarly, we can find out the frequency of other attributes.[Nel] This answers our question 2.5 in section 2.2.4.

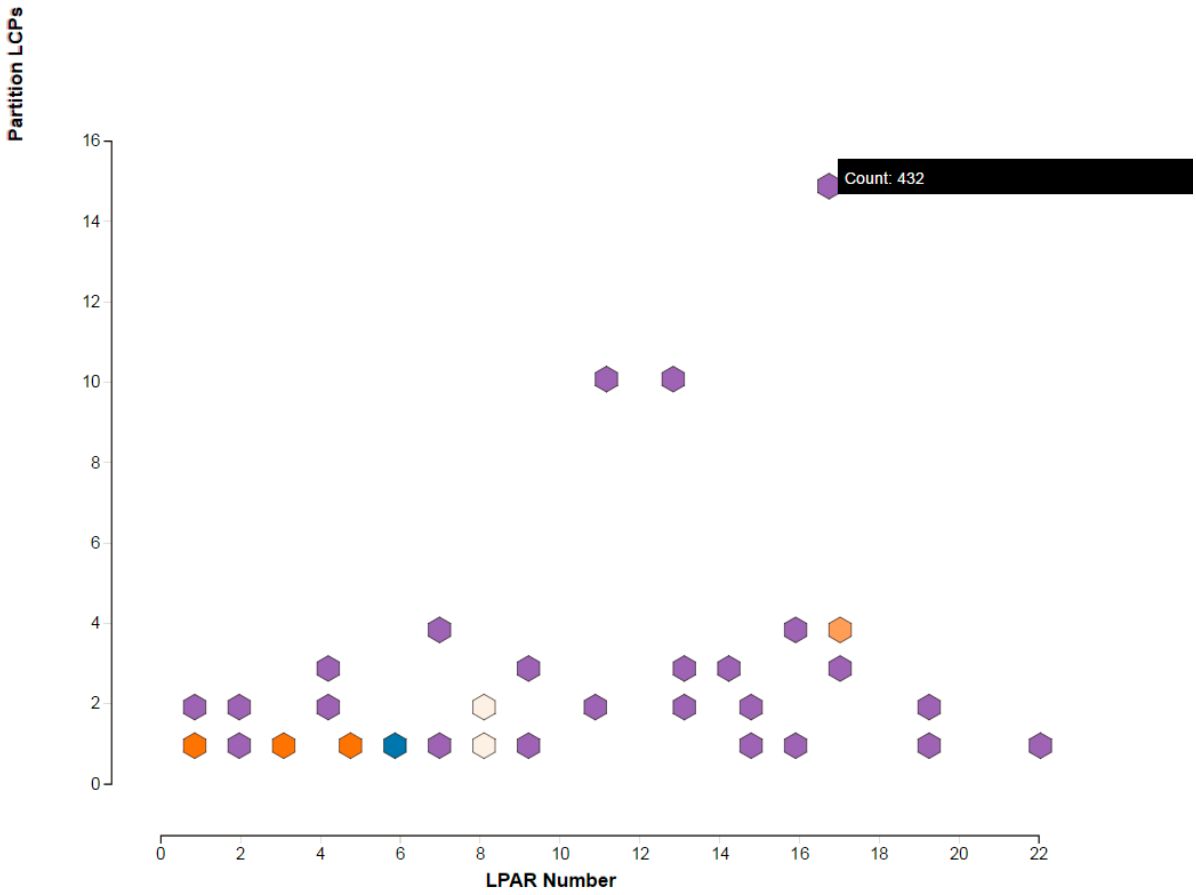

Figure 5.11: Frequency identification of emerging patterns between attributes Partition LCPs & LPAR Number in the dataset LFLTO7 through density plot

5.2.2.7 Multiline plot of S998DATA

S998DATA record contains 25 dimensions with 11250 observations. The multiline plot as shown in Figure 5.12 plots the x axis with time and y axis with ProTime and the 3rd dimension selected from panel as (named as Nested attributed in the panel for user selection) attribute Err is plotted with respect to time and ProTime dimensions.

You can observe that, Err contains two categorical values: Yes or No. Yes is represented with blue color and No is represented with orange color. This plot is helpful for

understanding the patterns by comparing one category with respect to the another. This answers our question 2.8 in section 2.2.4. However, the complexity of this diagram increases when the data record becomes huge. Hence, in later section, we look at the visual model that best suits for handling high dimensional datasets which makes user visual analysis much easier compared to the two dimensional plots. [AMM+08]

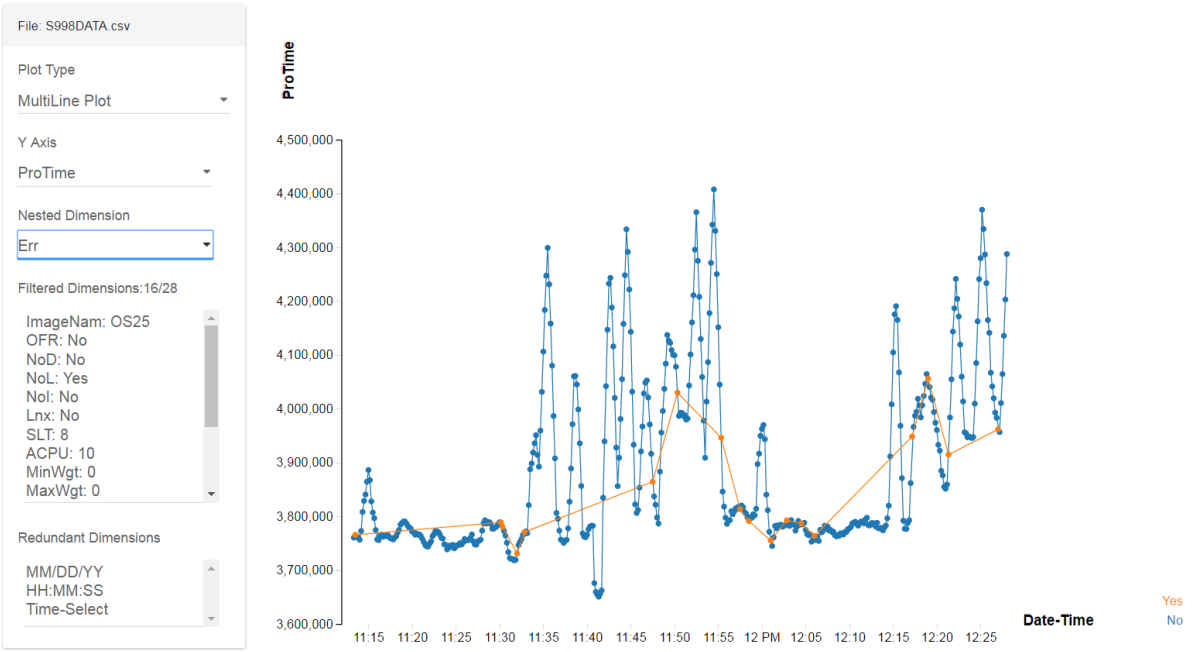

Figure 5.12: Nesting of attribute Err with ProTime of S998DATA in multiline plot

5.2.2.8 Parallel coordinates of S99CDATA

S99CDATA record contains 31 dimensions with 65968 data points. The high level view of all the attributes with in this record is shown in the Figure 5.13.

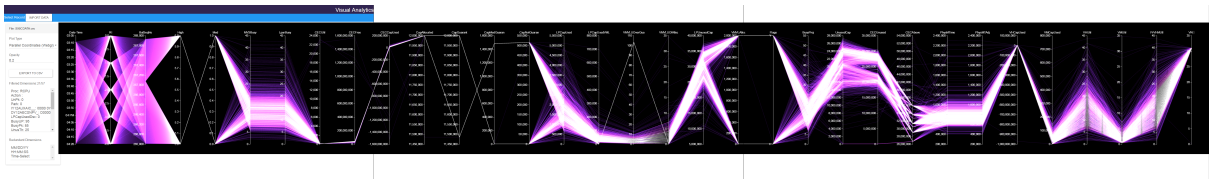

Figure 5.13: High level view of S99CDATA record in parallel coordinates domain

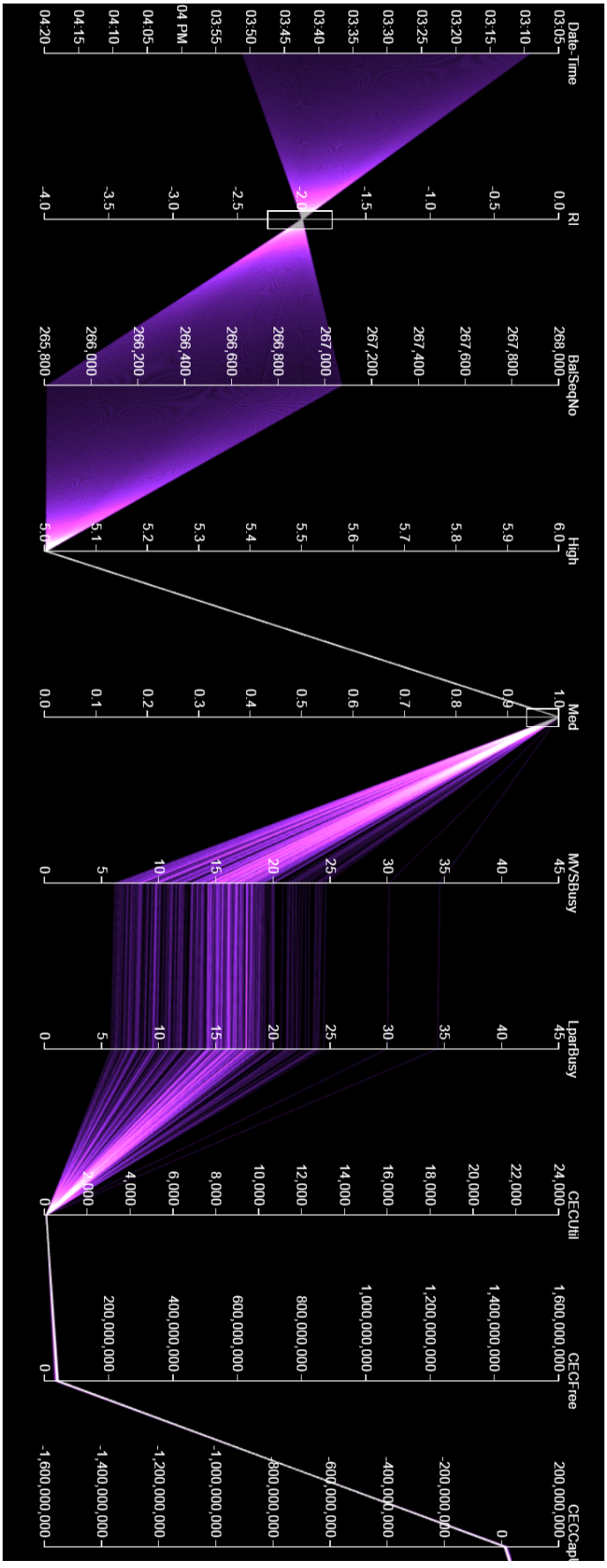

Figure 5.14: A sample snippet of brushed view of S99CDATA record on dimensions RI and Med

The paper "Comparing Interactive Web-Based Visualization Rendering techniques" [KSC12] states that visual analysis over web for high dimensional datasets gives better results with SVG overlay for user interactions and WebGL for rendering. This superimposing of techniques with different canvas layers is implemented in parallel coordinates as shown in Figure 5.13. The vertical axis layer is implemented with SVG for user interactions and additive blended polylines rendering with WebGL. [MM15]

Figure 5.13 shows us the high level view of S99CDATA record. By scrolling through the whole plot, you can easily identify which attributes are highly correlated, which are negatively correlated, outliers and emerging patterns. This answers our questions 2.2, 2.4, 2.7, 2.8 at once from section 2.2.4. In this dataset, attributes like High and Med are negatively correlated with cross over lines as stated in section 3.1.8. [HW13]

The detailed brushed view of the attributes can be seen in Figure 5.14. We performed multiple brush operations on S99CDATA record over attributes RI at -2.0 and Med at 1.0. The resulting filtered values are exported to CSV table as shown in Figure 5.15[HW13]

Date-Time	RI	BalSeqNo	High	Med	MVSBusy	LparBusy	CECUtil	CECFree	CECCapUs	CapAllocat	CapGuarar	CapMedG	CapNotGu
Wed May 10	-2	265806	5	1	13.31	13	20.7	37976795	10023205	12000000	11484375	1484375	515625
Wed May 10	-2	265816	5	1	12.18	11.87	19.14	38760518	9239482	12000000	11484375	1484375	515625
Wed May 10	-2	265821	5	1	11.87	11.57	17.96	39353933	8646067	12000000	11484375	1484375	515625
Wed May 10	-2	265826	5	1	12.93	12.63	21.48	37590849	10409151	12000000	11484375	1484375	515625
Wed May 10	-2	265831	5	1	12.43	12.09	20.7	38044725	9955275	12000000	11484375	1484375	515625
Wed May 10	-2	265836	5	1	14.5	14.26	25	35816616	12183384	12000000	11484375	1484375	515625
Wed May 10	-2	265866	5	1	14.43	14.21	23.82	36438397	11561603	12000000	11484375	1484375	515625
Wed May 10	-2	265876	5	1	12.81	12.4	20.7	37986360	10013640	12000000	11484375	1484375	515625
Wed May 10	-2	265881	5	1	15	14.82	21.48	37682653	10317347	12000000	11484375	1484375	515625
Wed May 10	-2	265891	5	1	13.87	13.65	19.92	38337459	9662541	12000000	11484375	1484375	515625

Figure 5.15: Exported brushed extractions of S99CDATA record to CSV table

Component	Functionality
Axis Layer	Scales of various attributes with labels
Polyline Layer	Connects all the data points over each axis
Interactions	Brushing and linking over multiple dimensions. Export the brushed values to CSV table. Optional change of transparency of rendered lines with the opacity control selection attached to the panel. Drag and Drop of dimensions adjacent to user interested attributes for comparison. Flipping the dimensions. Filtering the dimensions. Clustering through categories user selected.

Table 5.5: Functionality Description of Parallel Coordinates Components

5.2.2.9 Extended Features of Parallel Coordinates Plot

We shall consider S998DATA record for exploring various interaction features on parallel coordinates plot.

From Figure 5.16, we can observe that whole record is clustered according to the Err attribute categories. We can easily identify which cluster is dominating and taking more ProTime. This answers our questions 2.9, 2.10 and 2.11 from section 2.2.4.

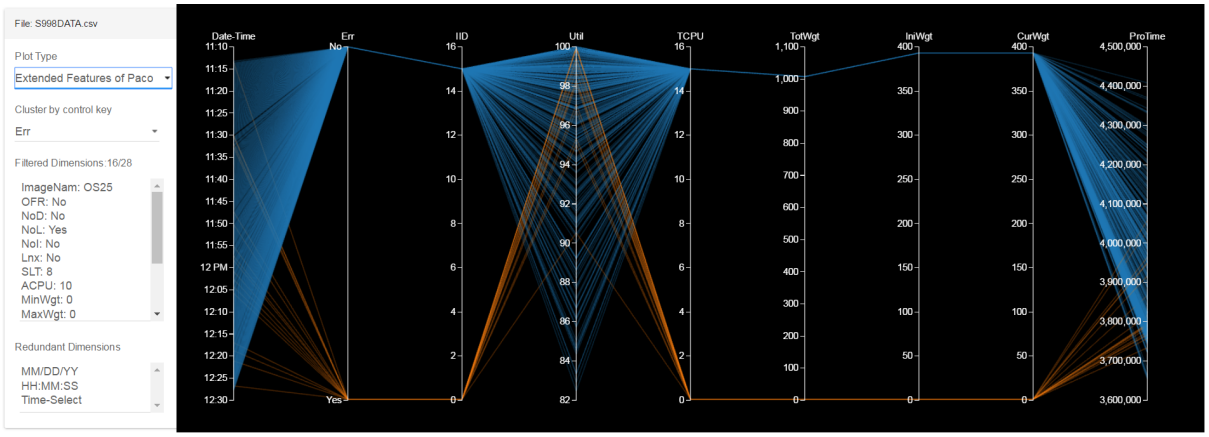

Figure 5.16: Cluster according to the user selected control key Err from the panel in S998DATA

There is also a correlation between various attributes like IID, TCPU and IniWgt, CurWgt. But, these attributes are not adjacent to clearly understand the relationship between them.

Hence, for identifying the patterns between various attributes which are not adjacent, we explore through drag and drop operation where user drags his/her interested attributes to the location user want to position for the analysis.

In Figure 5.17, we move the attributes IID and TCPU side by side and attributes IniWgt and CurWgt adjacent to one another. Also, we changed the positions of Util attribute near to ProTime and identifying the line crossings of these two attributes suggests they are negatively correlated.[GHWG14]

From Figure 5.17, we can also observe that, there is no need to maintain two copies of attributes which suggests same relation. Hence, we filter out the attributes IID and IniWgt by dragging the axis to the left side of the canvas. User need to click on the axis layer of the plot after all the filtering operations are performed to update the plot.

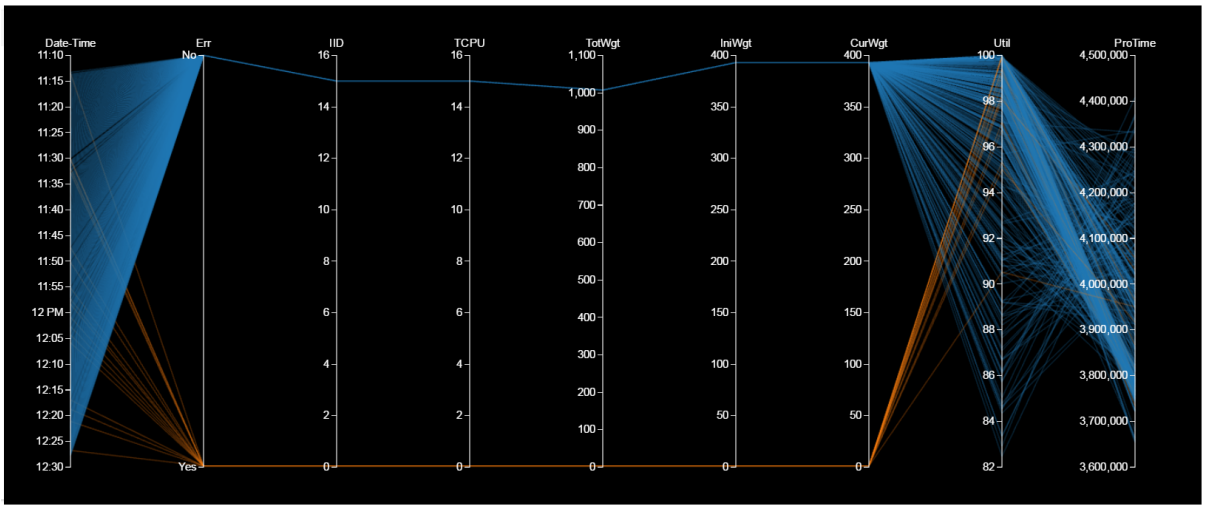

Figure 5.17: Changed dimension order in parallel coordinates plot

After performing the dimension filtering operation through visual analysis over the plot showed in Figure 5.17, the resulting plot can be observed in the Figure 5.18.[Ber].

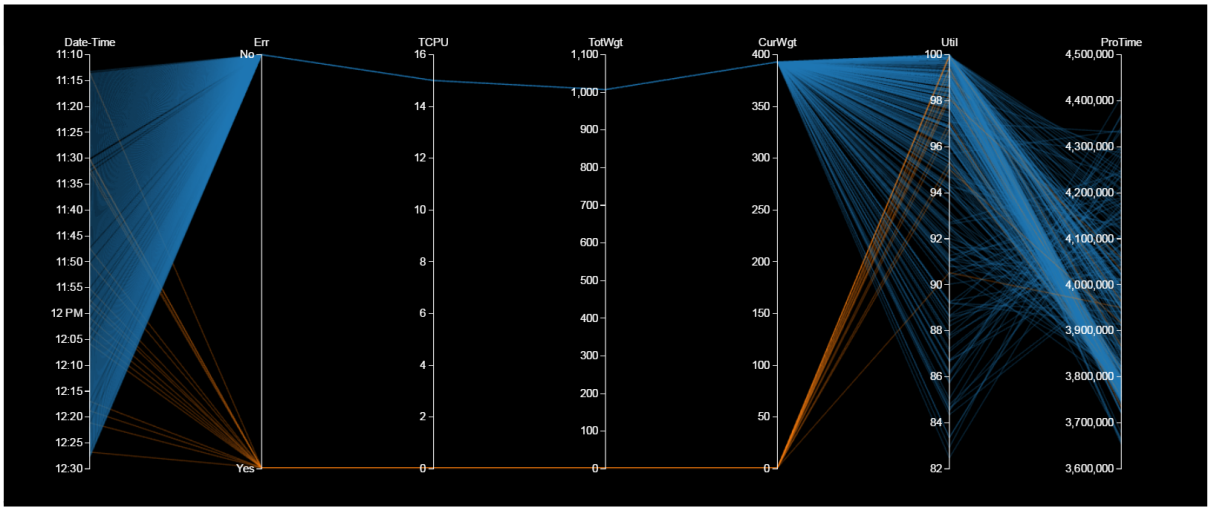

Figure 5.18: Filtered dimensions through user interaction in parallel coordinates plot

Further, we order the scale as specified in section 3.1.8 by flipping the axis upon user selection by clicking on the attribute name. This operation clears the clutter and user can identify the newly emerging patterns.

For instance, after flipping the ProTime attribute, the relation between attributes Util and ProTime is much clear. Similarly, flipping the Err attribute shows us at what time

interval category yes is distributed. We performed flipping over other attributes and the resulting plot can be observed in Figure 5.19.[HW13]

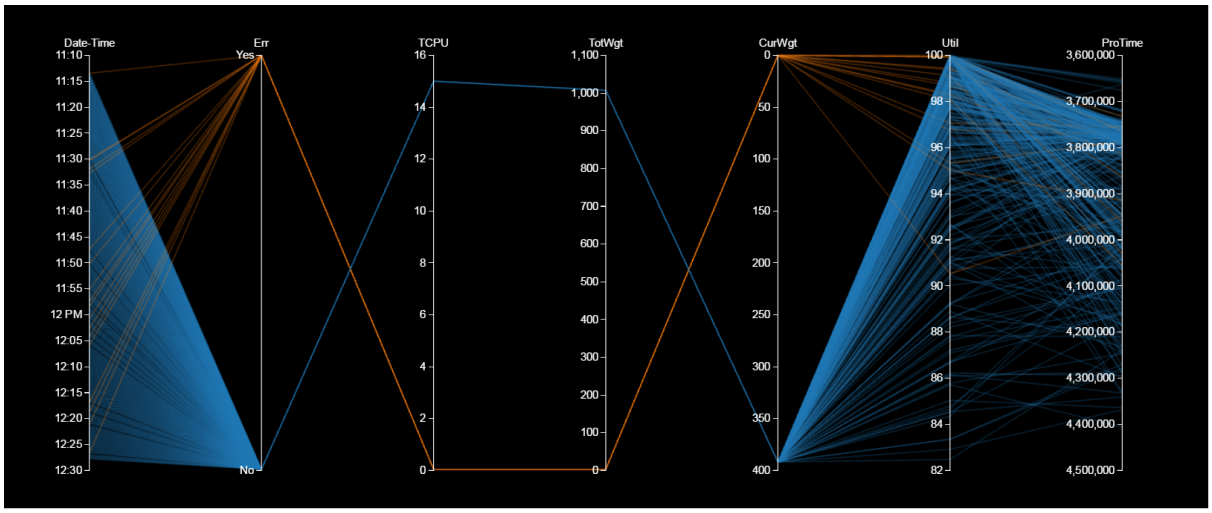

Figure 5.19: Flipped dimensions in parallel coordinates plot

5.2.2.10 Parallel Coordinates Plot with Analytical Support

User can choose the cluster number from panel and perform clustering according to the number of clusters. Later, the pre clustered observations are rendered to parallel coordinates plot.

This method of clustering is different from what we have seen in Figure 5.16. In Figure 5.16, user chooses the attribute which user want to cluster the whole dataset. The plot gets updated with their selection.

In K-means clustering, user passes the cluster number to algorithm and algorithm decides the random points and cluster accordingly as discussed in section 4.2.2.

From the plot shown in Figure 5.20, we can observe that user passed a cluster number of five and the record shows five different clusters. Each cluster is distinguished from one another by mapping data to the colors as we discussed in section 4.4.3 of colormaps.

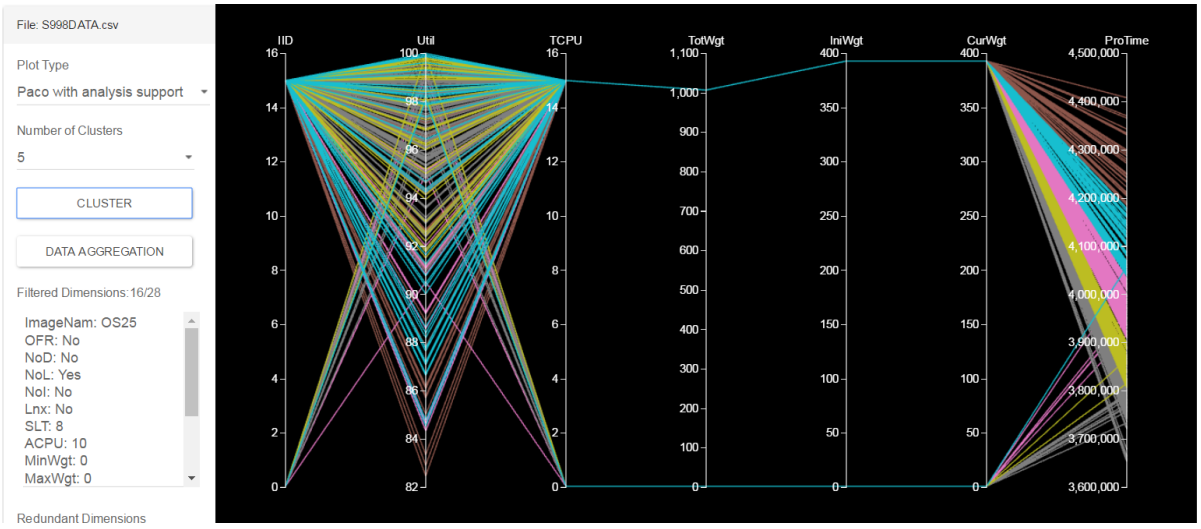

Figure 5.20: K-means clustering over the parallel coordinates plot

Data aggregation through means of the pre defined clusters gives us the summarized patterns as shown in Figure 5.21[HW13].

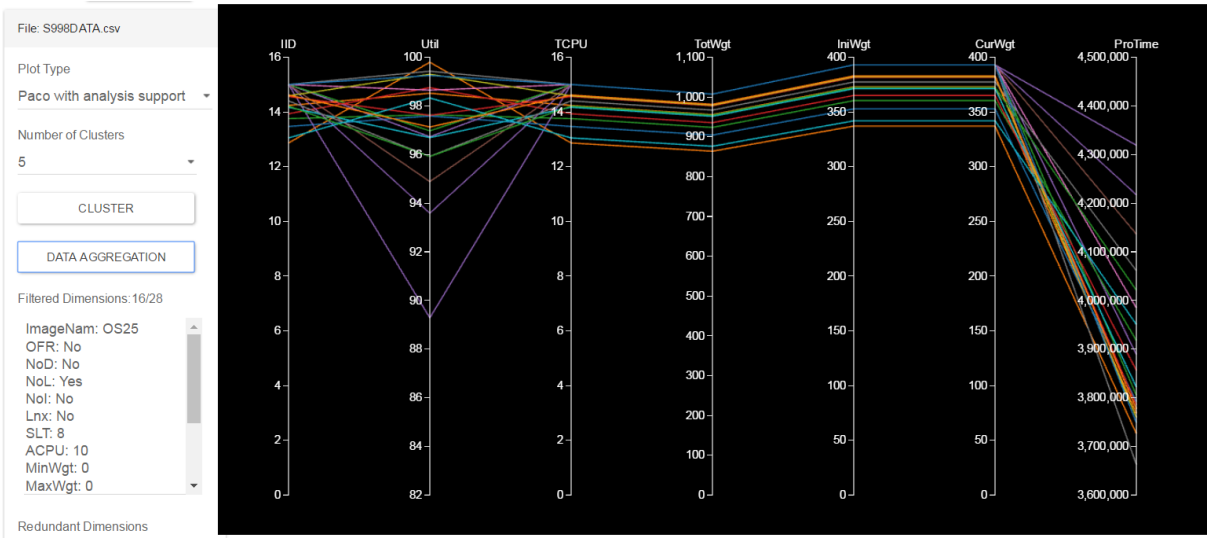

Figure 5.21: Data aggregation over the parallel coordinates plot

5.2.3 RMF Records and Visualization Scenarios

The sample snippet of SOUT723 is visualized through parallel coordinates plot as shown in the Figure 5.22. This record contains 70 dimensions with 19110 observations.

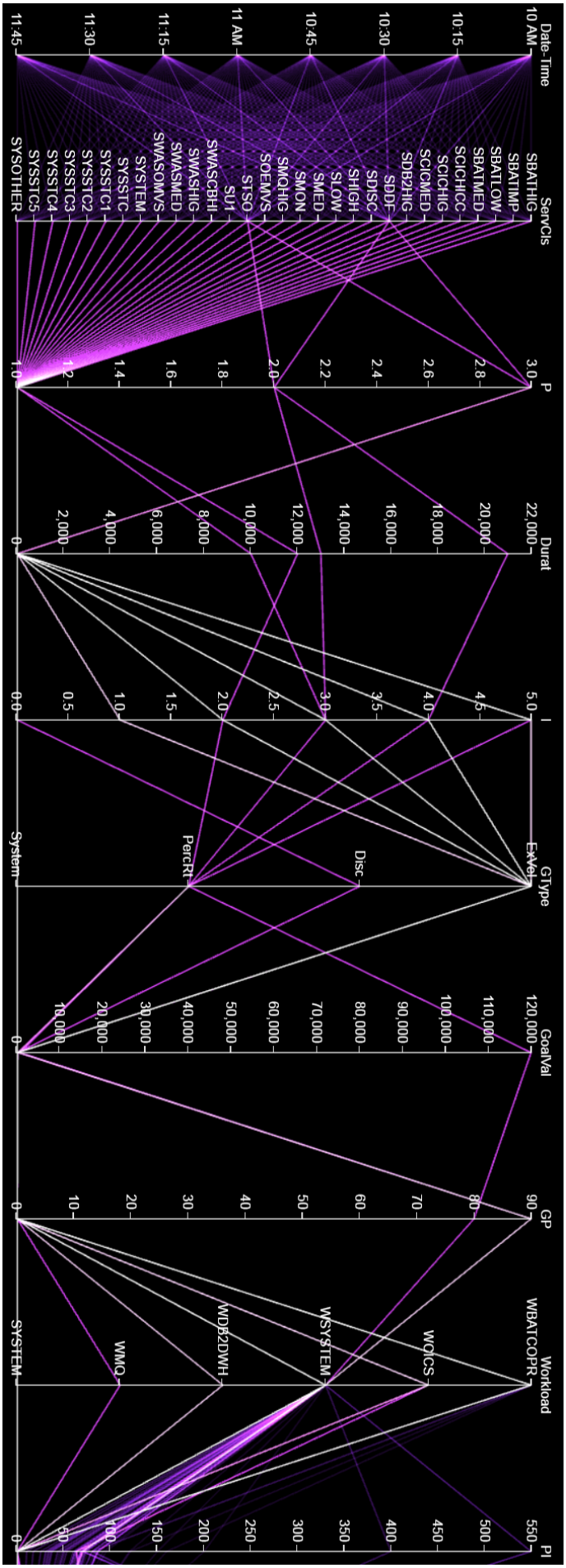

Figure 5.22: A sample snippet of SOUT723 RMF record

	Line	Multiline	Scatter	Dot	Heatmap	Density	SPLOM	Parallel Coordin
Interactions								
Link-Brush				X			X	X
Zoom				X				
Axis Change	X	X	X	X		X	X	
Axis Reorder							X	X
Flip Axis								X
Highlight				X			X	X
Mouse Hover		X		X	X	X		
Label change	X	X	X	X		X	X	
Filtering				X				X
Functionality								
Correlations			X	X		X	X	X
Clusters			X	X			X	X
Outliers	X		X	X			X	X
Patterns		X		X	X		X	X
Frequency					X			
Summarize						X		X
Aggregations					X			X
Trends		X						X
Distributions			X	X			X	X

Table 5.6: Summarized information of interactions and functionality of each visual model

The Table 5.6 gives us the summarized information of all the visual models and their interactions, functionality support.

5.3 Test Environment

The GPU rendering capacity for visualization of huge datasets depends on the graphics card we are using. Currently, we tested the models on NVIDIA GT-730m graphics card. Support for the rendering techniques we discussed in section 4.3 varies from browser to browser. We used chrome browser for most of our testing and development. The other essential details of the test laptop are listed in Table 5.7.

System Property	Model Details
Model	Lenovo ThinkPad T540p
Operating System	Windows 7 64-bit v1.14.00.AM B17
Graphics Card	NVIDIA GeForce-GT-730m
Processor	Intel Core i5-4300M CPU with 2.60GHz
RAM	15.7GB
Hard Disc	465GB
Display	LCD 1920x1080

Table 5.7: Details of the Test Laptop

6 User Case Study

We performed the user case study through three steps. At first, user training session is conducted to explain the usefulness of the visual models implemented and the applicability of the developed tool. Later, a task sheet is created explaining the exploration objectives for the user. At the end of exploration, we passed a survey sheet to understand the overall user experience.

A total of six experts are involved in the user case study. Three experts are from WLM team and three experts are from RMF team. Their details are discussed below:

- **Expert 1** is a software engineer at WLM Development team and industry supervisor for this thesis topic.
- **Expert 2** is a senior technical staff member of z/OS Design and Development.
- **Expert 3** is a senior technical staff member of z/OS Design and Development.
- **Expert 4** is a software support specialist for the z/OS RMF product. He is working at IBM since 30 years.
- **Expert 5** is a software engineer at RMF Development team.
- **Expert 6** is a software engineer with expertise in RMF Testing and Development.

Nine tasks are presented to each expert for their exploration as a part of this controlled experiment. They are:

- **Task 1** Familiarize with the tool
- **Task 2** Explore line plot to identify outliers
- **Task 3** Explore correlation heatmap for a summarized information of relation between attributes
- **Task 4** Explore scatter plot and understand clusters, correlations
- **Task 5** Explore dot plot for understanding the cluttered observations

- **Task 6** Explore scatter plot matrix by selecting three or more interested attributes for comparison of patterns and apply brushing technique to link the view in other plots.
- **Task 7** Explore density plot for frequency information of interested attributes
- **Task 8** Explore multiline plot for identifying attribute category trends
- **Task 9** Explore parallel coordinates plot through various interaction techniques incorporated in the plot like brushing and linking to select subset of items, drag and drop operation to compare adjacent attributes for correlations, filtering out the axis to remove unwanted attributes, flipping the axis to clear the clutter, exporting the brushed selections to CSV, clustering the plot through user selected attribute or by K-means clustering to find interesting patterns, aggregate the data for summarized information.

User training is given to all users at once and took around one hour for explaining the usefulness of various visual models and each of their significance. The time taken for individual exploration varied from each expert and their exploration objectives also differed. A brief description of their views and interests are noted below.

Experts 5 and 6 have taken a total time of an hour for exploration. They explored through all the visual models with the datasets provided and tested each of their functionality with respect to their goals.

At first, they started their exploration with S99CDATA and understood the filtered attributes which are not contributing for analysis. In line plot, they explored through CEC utilization and understood how it is changing with time. Later, they chose the capacity used attribute that is forming outliers at the negative values. From correlation heatmap, they understood that attribute MVSBusy is in high correlation with attribute LPARBusy. They checked in depth about these two attributes through scatter plot.

After this, they changed the dataset to S991DATA for further exploration with dot plot and they understood the distributions with in the selected attributes. Further by scatter plot matrix, they selected attributes Tot%, CPU%, ImpSyst. They understood from plot that Tot% and CPU% are in high linear correlation whereas when these two plots are compared with attribute ImpSyst, they showed no correlation.

Then they continued their exploration through density plot with attributes Imp3CPU, Imp4CPU to understand how the frequency is distributed among them. Further, they moved to multiline plot where third dimension is introduced to the 2D plot. Here, they understood the need for multivariate visual model to get information with no much visual complexity.

In parallel coordinates plot, they understood well the workload distribution when they are colored for comparison. They found that transaction response time and transaction execution time are in high correlation. Later, they exported the selected attributes to CSV. At the end, they chose five clusters and clustered through K-means and aggregated the items to find the interesting patterns. They understood the importance of visual models like parallel coordinates to derive information faster than other existing solutions like excel spreadsheets.

During the exploration, Experts 5 and 6 gave suggestions that users need to have the understanding of the records and the attributes of respective records. For instance, their exploration has to be supported with a detailed description of attributes for better understanding as specified in section 2.2.4 at Tables 2.1 and in appendix .

Experts 1 and 2 have taken total time of one hour together. All the models are explored with the new datasets one of the experts had. For instance, Expert 2 is interested to explore through S992DATA that has more than a million data points. Here, they experienced the fast rendering of parallel coordinates plot for rendering more data points at once. This improved their user experience to a large extent in guiding them to ask right questions from the visualization.

They selected one service class and checked how this service class is related to the other attributes. **Expert 3** have joined the session when experts 1 and 2 are continuing their exploration.

In the S992DATA record, they concentrated on the service class attribute. In particular, they brushed the attributes DB2, $SRMGOOD, $SRMBEST categories and checked how these categories are changing with respective to the time. This exploration says which workloads are getting distributed over particular time spans. They found some workloads are distributed over more time intervals and some are less frequent.

Further, they continued exploration by selecting the particular range in CPU delays from 90 to 100. They observed that they have very few discrete lines for this selection and those are repeating regularly for particular intervals of time. We understood that these explorations are crucial factor for making decisions after identifying the user's interest.

In the session with **Experts 1 and 4**, various visual models are explored to check the scalability and other interactive features. Here, expert 2 has joined the session and they composedly took a total time of an hour for their exploration.

Expert 4 suggested directly handling the raw data for visualizing the records. Currently, the records are processed from binary form with respective jobs specified in section 2.2.2. Experts 1, 2 showed further interest and suggested more ideas which we will discuss in the next chapter.

After the visual exploration with the tool presented, experts understood the important attributes and potential anomalies in less time. Currently, visual exploration through excel tool takes more time as user need to have domain specific knowledge of using those tools. These existing excel tools are visualizing records WLM, RMF separately with specific tools instead of providing single platform for visualization of all the SMF records.

Also, the excel tool has visual models that are focusing on cartesian coordinate system for visualization. As we can observe from the datasets presented in section five or the test data user tried in the case study, they have dimensions sometimes even ranging to 100. In this tool, we presented parallel coordinates visual model that is specific for this kind of multivariate data visualization to make the user exploration much easier and faster with interactive visual analysis.

A dynamic and scalable solution what users experienced through the tool we presented can significantly improve their decision making process to a greater extent.

Various questions about the tool are collected from these experts during and after their exploration. some of them are listed below:

- **Q1** Easier to derive important information from the data ?
- **Q2** Faster to get the summarized view of the records ?
- **Q3** Easy to understand the various visual models ?
- **Q4** Interaction mechanisms of visual models helpful for visual analysis ?
- **Q5** Understanding correlations between attributes became easier ?
- **Q6** Able to differentiate important attributes ?
- **Q7** Is user training sufficient ?
- **Q8** How would you rate the tool overall ?

For questions 1 to 7, we used likert scale of five values: strongly disagree, disagree, neutral, agree, strongly agree. For question 8, we used scale: excellent, good, satisfactory, fair, poor. These likert scales are mapped to numbers between 1 to 5 as shown:

- **Strongly Disagree/Poor** = 1
- **Disagree/Fair** = 2
- **Neutral/Satisfactory** = 3
- **Agree/Good** = 4

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Expert 1	5	5	4	5	5	5	5	5
Expert 2	5	4	1	5	5	4	3	5
Expert 3	4	5	4	5	5	3	4	5
Expert 4	5	4	2	4	4	3	4	4
Expert 5	4	5	3	5	4	4	4	5
Expert 6	3	4	4	4	4	4	5	5
Mean	4.4	4.5	3	4.7	4.5	3.9	4.2	4.9

Table 6.1: Expert answers for survey questions and mean of those values

- **Strongly Agree/Excellent** = 5

The questions and the expert answers are mapped from likert scales to the numerical values. Later, the mean of each question is calculated for all the expert answers. Those mean values lie between 3 to 4.9 as shown in the Table 6.1.

7 Conclusion and Future Work

This thesis developed a tool with visual models that support interactive visual analysis to make the exploration of SMF records easier for human cognition, faster with interactions, scalable to multiple dimensions and dynamic to explore different records.

The chapter two discussed briefly about information visualization model, scale types and visual encodings of discrete and continuous visualizations. These basics are applied over the SMF records for categorizing the data as time variant, understanding the scale type and meta data of various attributes along with identifying the potential quirks.

The chapter three introduced various visual models that best suit with data at hand like line plot, correlation heatmap, scatter plot, scatter plot matrix, dot plot, density plot, multiline plot and parallel coordinates.

The chapter four presented four stages of system design like data transformation, analysis, visualization and interaction before rendering the image to the user.

The chapter five introduced the production tools used in this project and presented the results after visual analysis of various SMF records. The questions that are raised during evaluation of records in chapter two are answered here through exploratory analysis.

The chapter six is about the user case study that is carried out with six potential users from WLM and RMF team. The user findings are monitored thought out their exploration and a survey is conducted at the end of their exploration for overall user experience.

During the case study, users interacted with various visual models and suggested an interaction feature in parallel coordinates plot. That is multiple brush operations on single axis layer. Currently, the plot supports brushing over different axes and unbrushing by clicking elsewhere on the axis.

However, to implement the feature user suggested, more emphasis lies with unbrushing. For instance, if users want to deselect one of the multiple brush operator on the axis, this undo operation has to be supported with special keyboard keys for each operator deselection or some other criteria. This interaction method needs further research.

Also, currently the parallel coordinates plot dimensions are increasing in direct proportion to the number of attributes. To maintain the context to the user screen display,

we can perform zooming operation similar to what we have implemented in dot plot. Instead of semantic zoom, we can try fisheye distortion. Upon mouse hover on the parallel coordinates plot, the user interested attributes are focused. This way user maintains the overall context. However, this model needs more user training and has to be designed with user requirements in mind.

Another interaction technique comes with heatmap. In this project, we concentrated on correlation heatmap. Correlation heatmap presents the correlation values upon user mouse hover on particular attribute pairs. This can be further extended to correlation density heatmap. It is combining correlation heatmap and density plot. Means, when user hovers on the attribute pairs in correlation heatmap, instead of displaying correlation values, we link a transparent layer with density plot. This way user gets the information about both frequency and correlation at same time.

The taxonomy of interaction techniques is ever growing with time. Interaction is always the top research challenge for analysis of multi variate visual models.

Appendices

Different types of processors within the mainframe environment as shown in Table A.1.

Processor	Description
CP	Central Processor or General purpose processor
IFL	Integrated Facility for Linux (under z/VM)
SAP	System Assist Processor- Handles various system accounting, management and I/O operations
zAAP	System z Application Assist Processor. Runs java and xml work
zIIP	System z Integrated Information Processor. Runs DB2, XML and IPSec workloads
ICF	Integrated Coupling Facility, supports parallel Sysplex operations

Table A.1: Types of Processors

Table A.2 describes about attributes of S99 Record for subtype 1.

Attribute	Description
Intv	Unique identifier for each record
MM/DD/YY	Date of the samples, format is Month-day-year
HH:MM:SS	Time in format, Hours-minutes-seconds
SID	Unique identifier of the system on which this record is created.
TOT%	Average utilization of all processors, regular CPs, zAAPs and SUPs/zIIPs
CPU%	Average utilization of regular CPs (processors)
IFA%	Average utilization of assist processors
#CP	Number of online processors
#IF	Number of online assist CPUs
IFANRM	Normalization factor for SUP processors

ImpSyst, Imp1CPU, Imp2CPU, Imp3CPU, Imp4CPU, Imp5CPU, ImpDisc, ImpFree	Each attribute contains the number of service units consumed by work in a regular CP at the corresponding importance level over the last ten seconds. The first entry contains service units with importance level zero, second with importance level one, and so on. The last entry contains service units that were not used in any service. The first level has the highest priority and this decreases down the line.
IFASYST, IMP1IFA, IMP2IFA, IMP3IFA, IMP4IFA, IMP5IFA, IFADISC, IFAFREE	Each attribute contains the number of service units consumed by work in an assist processor at the corresponding importance Level over the last ten seconds. The first entry contains service units in importance level zero, second with importance level one, and so on. The last entry contains service units that were not used in any service. The first level has the highest priority, this decreases with every level.
PgIns	Page-ins rate count used for calculating the system paging rate.
UIC1, UIC2, UIC3, UIC4	UIC bucket one is a count of frames that have most recently been referenced. Whereas, bucket four contains a count of frames that have not been referenced in a long time.
SHORT STAT SUP% #SP SUPNRM	Shortage flag Status of the flag Average utilization of SUP processors Number of online SUP processors Normalization factor for SUP processors
SUPSyst, Imp1SUP, Imp2SUP, Imp3SUP, Imp4SUP, Imp5SUP, SUPDisc, SUPFree,	Each field contains the number of service units consumed by work in SUP processor at the corresponding importance level over the last ten seconds. The first entry contains service units with importance level zero, second with importance level one, and so on. The last entry contains service units that were not used in any service. The first level has the highest priority, this decreases with every level
CPUDispTime CPUWaitTime CPUCapF CPUMaxF	Service units based on CPU dispatch time. Service units based on CPU wait time. Service units based on CPU capacity. Service units based on CPU maximum use.
IIPDispTime	Service units based on zIIPs dispatch time.

IIPWaitTime	Service units based on zIIPs wait time.
IIPCapF	Service units based on zIIPs capacity.
IIPMaxF	Service units based on zIIPs maximum use.
AAPDispTime	Service units based on zAAPs dispatch time.
AAPWaitTime	Service units based on zAAPs wait time.
AAPCapF	Service units based on zAAPs capacity.
AAPMaxF	Service units based on zAAPs maximum use.
LPARBusyTime	Logical partition's total busy time.
LPBsTimeAAP	Logical partition's time on zAAPs.
LPBsTimeIIP	Logical partition's time on zIIPs.
ZaaCpWai	Free zAAP LPAR capacity based on the accumulated logical zAAP wait times.
ZaaCpGra	Free zAAP LPAR capacity based on the zAAP LPAR weight.
ZaaCpCEC	Free zAAP LPAR capacity which is the total of what is always available to the LPAR and the portion of the unused zAAP capacity of the CEC.
ZaaCpLCP	Free zAAP LPAR capacity based on the configured Logical zAAPs.
ZaaFreCp	Total free zAAPs capacity.
ZiiCpWait	Free zIIP LPAR capacity based on the accumulated logical zIIP wait times.
ZiiCpGra	Free zIIP LPAR capacity based on the zIIP LPAR weight.
ZiiCpCEC	Free zIIP LPAR capacity which is the total of what is always available to the LPAR and the portion of the unused zIIP capacity of the CEC.
ZiiCpLCP	Free zIIP LPAR capacity based on the configured logical zIIPs.
ZiiFreCp	Total free zIIPs capacity.

Table A.2: S991DATA Attributes Description

Glossary

Abstract data Abstract data refers to (heterogeneous) data with no inherent spatial structure; It does not allow for a straightforward mapping to any geometry, but relies upon means provided by information visualization for its visual representation[wikd].

Linting Linting is a program that ensures quality of code and checks for errors in it.

Saturation Saturation defines how pure colors are to user. A high-saturation color is vivid, and a low-saturation color is close to black, white, or gray.[War12].

Service Unit It is used to measure consumption of resources by the workload.

Sysplex IBM mainframes introduced the term Sysplex which means systems complex in MVS. Sysplex makes components in upto eight LPARs to communicate with each other.

Unit Testing Unit testing tests the individual modules of a software and validates every module is behaving as designed.

List of Abbreviations

CICS Customer Information Control System.

DIAG Diagnostic information. It is a return code.

DOM Document Object Model.

InfoVis Information Visualization.

LPAR Logical Partition.

MSU Million Service Unit.

NPM Node Package Manager.

OLTP Online Transaction Processing.

OpenGL ES Open Graphics Library for Embedded Systems.

Rexx Restructured Extended Executor.

RMF Resource Measurement Facility.

SciVis Scientific Visualization.

SMF System Management Facility.

SPLOM Scatter Plot Matrix.

SVG Scalable Vector Graphics.

TSO Time Sharing Option.

WebGL Web Graphics Library.

WLM Work Load Management.

Bibliography

[AMM+08] W. Aigner, S. Miksch, W. Muller, H. Schumann, C. Tominski. In: *Visualization and Computer Graphics, IEEE Transactions on* 14 (2008). DOI: 10.1109/TVCG.2007.70415. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4359494 (cit. on pp. 36, 72).

[AMST11] W. Aigner, S. Miksch, H. Schumann, C. Tominski. *Visualization of Time-Oriented Data*. 1st. Springer Publishing Company, Incorporated, 2011. ISBN: 0857290789, 9780857290786 (cit. on pp. 22, 23, 35).

[BAB+93] D. M. Butler, J. C. Almond, R. D. Bergeron, K. W. Brodlie, R. B. Haber. "Visualization Reference Models." In: *Proceedings of the 4th Conference on Visualization '93*. VIS '93. San Jose, California: IEEE Computer Society, 1993, pp. 337–342. ISBN: 0-8186-3940-7. URL: http://dl.acm.org/citation.cfm?id=949845.949906 (cit. on p. 25).

[BBH+17] M. Behrisch, B. Bach, M. Hund, M. Delz, L. V. Ruden, J.-D. Fekete, T. Schreck. "Magnostics: Image-Based Search of Interesting Matrix Views for Guided Network Exploration." In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 31–40. ISSN: 1077-2626. DOI: doi.ieeecomputersociety.org/10.1109/TVCG.2016.2598467 (cit. on pp. 42, 43, 58).

[BBW16] F. Beck, M. Burch, D. Weiskopf. "A Matrix-Based Visual Comparison of Time Series Sports Data." In: *Vision, Modeling and Visualization*. Ed. by M. Hullin, M. Stamminger, T. Weinkauf. The Eurographics Association, 2016. ISBN: 978-3-03868-025-3. DOI: 10.2312/vmv.20161342 (cit. on pp. 40, 43).

[BC87] R. A. Becker, W. S. Cleveland. "Brushing Scatterplots." In: *Technometrics* 29.2 (May 1987), pp. 127–142. ISSN: 0040-1706. DOI: 10.2307/1269768. URL: http://dx.doi.org/10.2307/1269768 (cit. on pp. 43, 56, 68).

[Ber] ischool Berkeley. *Brushing in Parallel Coordinates*. URL: http://people.ischool.berkeley.edu/~japple/jeopardy/ (cit. on pp. 47, 76).

[Ber81] J. Bertin. *Graphics and Graphic Information-processing*. de Gruyter, 1981. ISBN: 9783110088687. URL: https://books.google.de/books?id=2tlQAAAAMAAJ (cit. on pp. 22, 25).

[Boo] I. Books. *Attributes Description of Various Records*. URL: https://www.ibm.com/support/knowledgecenter/SSLTBW_2.2.0/com.ibm.zos.v2r2.ieag200/toc.htm (cit. on pp. 28, 32).

[Bos] M. Bostock. *Data Driven Documents*. URL: https://d3js.org/.

[bru] brunch.org. *Brunch Build Tool*. URL: http://brunch.io/docs/config (cit. on p. 61).

[CLM13] D. Carr, N. Lewin-Koh, M. Maechler. *hexbin: Hexagonal Binning Routines*. R package version 1.26.2. 2013. URL: http://CRAN.R-project.org/package=hexbin (cit. on p. 38).

[CMS99] S. K. Card, J. D. Mackinlay, B. Shneiderman, eds. *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999. ISBN: 1-55860-533-9.

[cou] coursera.org. *Tidying Dataset*. URL: https://www.coursera.org/learn/data-scientists-tools (cit. on p. 49).

[CUa] R. Cabello, B. Ulicny. *Three.js*. URL: https://threejs.org/.

[CUb] M. H. Cynthia Brewer, T. P. S. University. *Colorbrewer*. URL: http://www.colorbrewer2.org, (cit. on p. 57).

[dat] datavizcatalogue. *Correlation Strength*. URL: http://datavizcatalogue.com/methods/line_graph.html.

[dem] de.mathworks.com. *K-means*. URL: https://de.mathworks.com/help/stats/kmeans.html?requestedDomain=www.mathworks.com (cit. on p. 51).

[Dir14] J. Dirksen. *Three.js Essentials*. Community experience distilled. Packt Publishing, 2014. ISBN: 9781783980871. URL: https://books.google.de/books?id=3mT4AwAAQBAJ.

[EKO+12] M. Ebbers, J. Kettner, W. O'Brien, B. Ogden, I. Redbooks. *Introduction to the New Mainframe: z/OS Basics*. IBM Redbooks, 2012. ISBN: 9780738435343. URL: https://books.google.de/books?id=c-a1AgAAQBAJ (cit. on pp. 15, 16).

[ESBB98] M. B. Eisen, P. T. Spellman, P. O. Brown, D. Botstein. "Cluster analysis and display of genome-wide expression patterns." In: *Proceedings of the National Academy of Sciences* 95.25 (1998), pp. 14863–14868. eprint: http://www.pnas.org/cgi/reprint/95/25/14863.pdf. URL: http://www.pnas.org/cgi/content/abstract/95/25/14863 (cit. on p. 41).

[fil] fileformat.info. *Encoding Algorithm*. URL: http://www.fileformat.info/mirror/egff/ch09_03.htm.

[FWG02] U. Fayyad, A. Wierse, G. Grinstein. *Information Visualization in Data Mining and Knowledge Discovery*. The Morgan Kaufmann series in data management systems. Morgan Kaufmann, 2002. ISBN: 9781558606890. URL: https://books.google.de/books?id=rYFvnyPRwkgC (cit. on pp. 38, 62, 68).

[GHWG14] S. Grottel, J. Heinrich, D. Weiskopf, S. Gumhold. "Visual Analysis of Trajectories in Multi-Dimensional State Spaces." In: *Comput. Graph. Forum* 33.6 (Sept. 2014), pp. 310–321. ISSN: 0167-7055. DOI: 10.1111/cgf.12352. URL: https://doi.org/10.1111/cgf.12352 (cit. on pp. 44, 75).

[Hav15] J. Havill. *Discovering Computer Science: Interdisciplinary Problems, Principles, and Python Programming*. Chapman and Hall/CRC, 2015. ISBN: 148225414X, 9781482254143.

[HB] J. Heinrich, B. Broeksema. "Big Data Visual Analytics with Parallel Coordinates." In: (). DOI: 10.1109/BDVA.2015.7314286. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7314286&isnumber=7314277 (cit. on p. 45).

[Hof00] P. E. Hoffman. "Table Visualizations: A Formal Model and Its Applications." AAI9950455. PhD thesis. 2000. ISBN: 0-599-54796-0 (cit. on p. 25).

[HS15] W. Härdle, L. Simar. *Applied Multivariate Statistical Analysis*. Springer Berlin Heidelberg, 2015. ISBN: 9783662451717. URL: https://books.google.de/books?id=KI3dBgAAQBAJ.

[HSW12] J. Heinrich, J. Stasko, D. Weiskopf. "The Parallel Coordinates Matrix." In: *EuroVis - Short Papers*. Ed. by M. Meyer, T. Weinkaufs. The Eurographics Association, 2012. ISBN: 978-3-905673-91-3. DOI: 10.2312/PE/EuroVisShort/EuroVisShort2012/037-041 (cit. on p. 45).

[Hua13] M. Huang. *Innovative Approaches of Data Visualization and Visual Analytics*. Advances in Data Mining and Database Management: IGI Global, 2013. ISBN: 9781466643109. URL: https://books.google.de/books?id=SNuWBQAAQBAJ (cit. on p. 51).

[HW09] J. Heinrich, D. Weiskopf. "Continuous Parallel Coordinates." In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (Nov. 2009), pp. 1531–1538. ISSN: 1077-2626. DOI: 10.1109/TVCG.2009.131. URL: http://dx.doi.org/10.1109/TVCG.2009.131.

[HW13] J. Heinrich, D. Weiskopf. "State of the Art of Parallel Coordinates." In: *Eurographics 2013 - State of the Art Reports*. Ed. by M. Sbert, L. Szirmay-Kalos. The Eurographics Association, 2013. DOI: 10.2312/conf/EG2013/stars/095-116 (cit. on pp. 44, 46, 52, 56, 74, 77, 78).

[IS11] N. Iliinsky, J. Steele. *Designing Data Visualizations: Representing Informational Relationships*. O'Reilly Media, 2011. ISBN: 9781449317065. URL: https://books.google.de/books?id=ygLH4qrGm-wC (cit. on p. 58).

[jso] json.org. *JSON*. URL: http://www.json.org (cit. on p. 49).

[Kei] D. A. Keim. "Information Visualization and Visual Data Mining." In: ().

[Kei01] D. A. Keim. "Visual Exploration of Large Data Sets." In: *Commun. ACM* 44.8 (Aug. 2001), pp. 38–44. ISSN: 0001-0782. DOI: 10.1145/381641.381656. URL: http://doi.acm.org/10.1145/381641.381656 (cit. on p. 34).

[KMN+] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, A. Y. Wu. "An Efficient k-Means Clustering Algorithm: Analysis and Implementation." In: *IEEE Trans. Pattern Anal. Mach. Intell.* (). URL: http://dx.doi.org/10.1109/TPAMI.2002.1017616 (cit. on p. 51).

[Kna15] C. Knaflic. *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Wiley, 2015. ISBN: 9781119002062. URL: https://books.google.de/books?id=rRSRCgAAQBAJ (cit. on p. 41).

[Koc] D. S. Koch. *Information visualization and visual analytics classes at university of stuttgart*. URL: https://www.vis.uni-stuttgart.de/nc/lehre/details/typ/vorlesung/2666/267.html (cit. on pp. 43, 45).

[KSC12] D. E. Kee, L. Salowitz, R. Chang. "Comparing Interactive Web-Based Visualization Rendering Techniques." In: 2012 (cit. on p. 74).

[Mac86] J. Mackinlay. "Automating the Design of Graphical Presentations of Relational Information." In: *ACM Trans. Graph.* 5.2 (Apr. 1986), pp. 110–141. ISSN: 0730-0301. DOI: 10.1145/22949.22950. URL: http://doi.acm.org/10.1145/22949.22950 (cit. on p. 25).

[mat] mathisfun.com. *Data Correlations*. URL: https://www.mathsisfun.com/data/correlation.html (cit. on p. 38).

[MM15] T. Munzner, E. Maguire. *Visualization analysis and design*. AK Peters visualization series. Boca Raton, FL: CRC Press, 2015. URL: https://cds.cern.ch/record/2001992 (cit. on pp. 23–26, 37, 38, 42, 46, 50, 52, 57, 69, 74).

[Mur13] S. Murray. *Interactive Data Visualization for the Web*. O'Reilly Media, Inc., 2013. ISBN: 1449339735, 9781449339739.

[Nag06] H. R. Nagel. "Scientific Visualization versus Information Visualization." In: 2006 (cit. on p. 27).

[Nel] F. Nelli. *Density Plot*. URL: http://www.meccanismocomplesso.org/en/hexagonal-binning/ (cit. on pp. 39, 40, 71).

[nod] nodejs.org. *Node Server*. URL: https://nodejs.org/en/ (cit. on p. 59).

[Ofu14] J. Ofungwu. *Statistical applications for environmental analysis and risk assessment*. John Wiley and Sons, 2014.

[Øye15] V. Øye. "Accelerating nonlinear image transformations with OpenGL ES: A study on fish-eye undistortion." MA thesis. 2015 (cit. on p. 54).

[PAE+08] L. Parziale, E. Alves, K. Egeler, C. Jordan, J. Herne, E. Dow, E. Naveen, M. Pattabhiraman, K. Smith, I. Redbooks. *Introduction to the New Mainframe: z/VM Basics*. IBM Redbooks, 2008. ISBN: 9780738488554. URL: https://books.google.de/books?id=5ee1AgAAQBAJ.

[Red05] I. Redbooks. *Effective zSeries Performance Monitoring Using Resource Measurement Facility*. IBM redbooks. IBM, International Technical Support Organization, 2005. ISBN: 9780738492353. URL: https://www.redbooks.ibm.com/abstracts/sg246645.html (cit. on p. 19).

[Red07] I. Redbooks. *System Programmer's Guide to Workload Manager*. IBM redbooks. IBM, International Technical Support Organization, 2007. ISBN: 9780738489933. URL: https://books.google.de/books?id=Js8sPQAACAAJ (cit. on pp. 17, 18).

[RFP15] J. Resig, R. Ferguson, J. Paxton. *Pro JavaScript Techniques: Second Edition*. 2nd. Berkely, CA, USA: Apress, 2015. ISBN: 1430263911, 9781430263913 (cit. on pp. 59–61).

[RSM+16] S. Radoš, R. Splechtna, K. Matkovic, M. Duras, E. Gröller, H. Hauser. "Towards Quantitative Visual Analytics with Structured Brushing and Linked Statistics." In: *Computer Graphics Forum* (2016). ISSN: 1467-8659. DOI: 10.1111/cgf.12901 (cit. on pp. 46, 56).

[sci] scikit. URL: http://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation (cit. on p. 51).

[Ste46] S. S. Stevens. "On the Theory of Scales of Measurement." In: *Science* 103.2684 (1946), pp. 677–680 (cit. on p. 26).

[TM02] M. Tory, T. Moeller. *A Model-Based Visualization Taxonomy*. 2002 (cit. on pp. 27, 28).

[tut] tutorialspoint. *WebGL*. URL: http://www.tutorialspoint.com/webgl/ (cit. on pp. 53–55).

[Vau13] R. Vaupel. *High Availability and Scalability of Mainframe Environments using System z and z/OS as example*. KIT Scientific Publishing, 2013 (cit. on pp. 16, 18).

[War12] C. Ware. *Information Visualization: Perception for Design*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012. ISBN: 9780123814647, 9780123814654 (cit. on pp. 38, 93).

[WB97] P. C. Wong, R. D. Bergeron. "30 Years of Multidimensional Multivariate Visualization." In: *Scientific Visualization, Overviews, Methodologies, and Techniques*. Washington, DC, USA: IEEE Computer Society, 1997, pp. 3–33. ISBN: 0-8186-7777-5. URL: http://dl.acm.org/citation.cfm?id=647365.725689 (cit. on pp. 37, 42, 44).

[WDH+17] B. White, C. De Leon, E. Hoogerbrug, E. Palacio, F. Pinto, B. Sannerud, M. Soellig, J. Troy, J. Yang, I. Redbooks. *IBM z13 and IBM z13s Technical Introduction*. IBM Redbooks, 2017. ISBN: 9780738441603. URL: https://books.google.de/books?id=tT-2CwAAQBAJ.

[WGK10] M. Ward, G. Grinstein, D. Keim. *Interactive Data Visualization: Foundations, Techniques, and Applications*. Natick, MA, USA: A. K. Peters, Ltd., 2010. ISBN: 1568814739, 9781568814735 (cit. on pp. 23, 27, 34, 41, 42, 57).

[Wik] Wikipedia. *Computer Graphics*. URL: https://en.wikipedia.org/wiki/Computer_graphics (cit. on pp. 52, 53).

[wika] wiki. *K-means*. URL: https://en.wikipedia.org/wiki/K-means_clustering (cit. on p. 51).

[wikb] wiki. *Lineplot*. URL: http://https://en.wikipedia.org/wiki/Line_chart.

[wikc] wiki. *REXX for SMF Extractions*. URL: https://en.wikibooks.org/wiki/SMF_Records/How_to_Extract_Values_from_SMF_Record_Fields (cit. on p. 29).

[wikd] I. wiki. *Abstract data definition*. URL: http://www.infovis-wiki.net/index.php?title=Abstract_data (cit. on p. 93).

[wike] I. wiki. *Visualization Pipeline*. URL: http://www.infovis-wiki.net/index.php?title=File:Dossantos04vis_pipeline.png (cit. on p. 22).

[wikf] I. wiki. *Zoom Types*. URL: http://www.infovis-wiki.net/ (cit. on pp. 56, 57).

[Wor11] Workshop. *Visual Analytics Workshop*. 2011. URL: http://infovis.cs.vt.edu/sites/default/files/p33-north.pdf (cit. on p. 22).

[WWR+05] L. Wilkinson, D. Wills, D. Rope, A. Norton, R. Dubbs. *The Grammar of Graphics*. Statistics and Computing. Springer New York, 2005. ISBN: 9780387245447. URL: https://books.google.de/books?id=%5C_kRX4LoFfGQC (cit. on p. 25).

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature