Paper 278-2012

Hyperslicing with SAS

Jonathon Khoo Kay Chin, School of Information System, SMU, Singapore

Koo Ping Shung, School of Information System, SMU, Singapore

ABSTRACT

Hyperslice is one of the newer techniques in visualization that involves the viewing of high dimensional data in a simple visualization. The data is centered around a chosen point based on a certain dimension, creating a slice. The user can further explore this data through manipulating this chosen point. However, as SAS does not provide this type of interactivity directly, in this paper, we explore some possible ways to implement this extreme slicing of data and exploration of the information using the same concepts proposed in Hyperslice within different aspects of SAS.

INTRODUCTION

Visualizing high dimensional data has always been an issue in data exploration as the human mind has difficulty comprehending data with N > 3 dimensions. Traditional visualizations, such as 3D Scatter plots, are only able to handle up to 4 dimensions at best while still being humanly understood. This limits the ability of users to explore the data and understand how the different variables interact with one another simultaneously. This issue has been solved with the development of the hyperslice visualization technique.

As SAS does not inherently have hyperslice support, we will be looking at some possible implementations of this technique using PROC SGSCATTER, AF Frames, and JMP.

HYPERSLICE

Hyperslice was developed by Jarke van Wijk and Robert van Liere [1] as a new technique to visualize scalar functions across multiple dimensions. The key underlying premise is that the combination of visualization, representation and interaction are all equally important and they are extremely closely related for proper understanding of the data. The understanding of the data does not stop at the geometric level and should extend to the behavioral level to improve the overall understanding of the concept and idea.

Normally, to effectively visualize relationships, the number of dimensions is reduced to two to four. This, however, limits greatly our understanding, only providing us a glimpse of the data. Thus, by using scatterplot matrices or trellis plots, we can bring out a more complete picture of the data. Hyperslice brings this one step further by extracting the dependent variable and using it as the point of interaction to gain insights across all the corresponding plots. In technical terms, Hyperslice is a compilation of orthogonal pair wise plots of the variables with the single ability of varying according to the values of one particular variable.

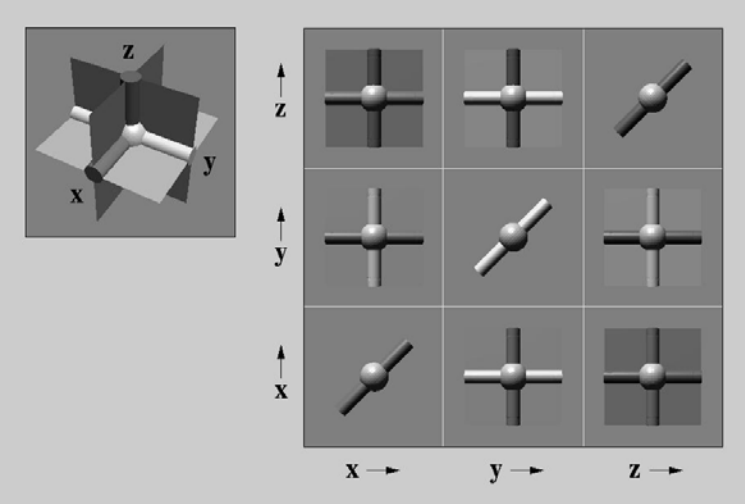

Figure 1: Hyperslice concept.

To bring this concept into SAS, we experimented with different products and components to see how much we would be able to implement.

For BASE SAS, one of the key issues we faced was the inability to generate graphs with interactivity. To overcome this problem, we will be using a simple trick in SAS to emulate as far as possible the ability to create something that looks interactive. To create this plot, we will be using PROC SGSCATTER.

PROC SGSCATTER

PROC SGSCATTER is one of the new graph plotting PROCs that is now in SAS 9.2. PROC SGSCATTER has several interesting plots that it can generate easily. Let us have a look at the syntax of PROC SGSCATTER.

```
PROC SGSCATTER < options>;

    COMPARE X= variable | (variable-1 ... variable-n) Y= variable | (variable-1 ...
    variable-n)</options>;

    MATRIX variable-1 variable-2 < ... variable-n > </options>;

    PLOT plot-request(s) </options>;
```

PROC SGSCATTER can produce different plots by manipulating the compare and matrix statements. Scatter plot matrix can be created with the following statements.

```
proc sgscatter data=sashelp.iris;
  title "Scatterplot Matrix for Iris Data";
  matrix sepallength petallength sepalwidth petalwidth
         / diagonal = (histogram) group=species;
run;
```

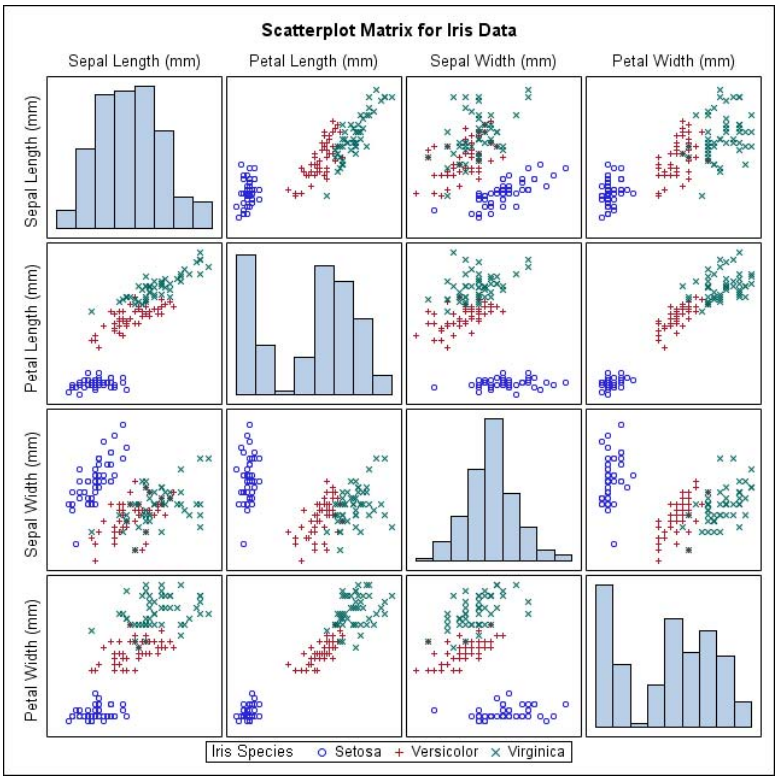

Figure 2: Scatterplot Matrix

Trellis plots or comparative plots can be easily done with the codes below.

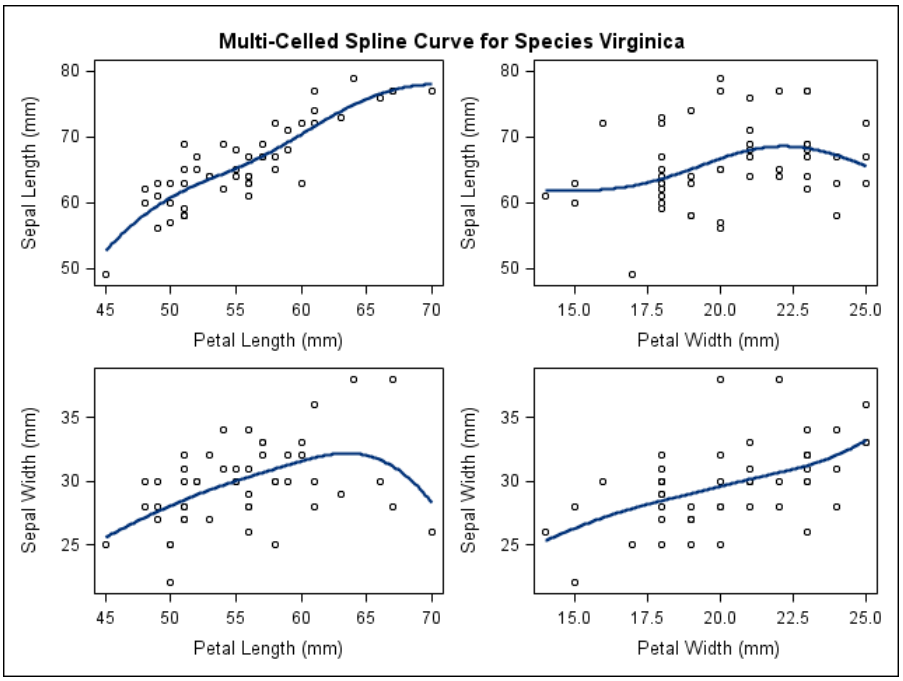

Figure 3: Comparative Trellis Plots

We can observe that PROC SGSCATTER has many capabilities that can be exploited to do multiple variables comparison. However, in the case of hyperslice, we have to incorporate a third moving dimension into the graph. Most fortunately, we can do this with a little trick in SAS and SGSCATTER.

Since hyperslice attempts to highlight the data as we move through a variable acting as the third dimension, we can easily code it into the data to highlight points as we move through the data. At the same time, as opposed to highlighting the data at every single point of the third dimension, we will be using a ranked order approach. This approach allows us to visualize movement of data across suitable range of values as opposed to looking at item point by point.

APPROACH 1: HIGHLIGHTING

To be able to visualize the movement through the variable, we will have to plot many such charts to see the changes. The macro below creates the same chart by highlighting the different sections of the data with each run.

```
/*HYPERSLICE*/

%MACRO HYPERSLICE(DATA,VAR,GROUP,LEVELS);

      PROC SORT DATA = &DATA;BY &GROUP;RUN;

      PROC RANK DATA = &DATA(KEEP = &VAR &GROUP) OUT = TEMP GROUPS = &LEVELS;
      VAR &GROUP;RANKS &GROUP.R;
      RUN;

      DATA TEMP;
      SET TEMP;
      %DO I = 0 %TO %EVAL(&LEVELS-1);
            IF &GROUP.R = &I THEN L&I = 1;ELSE L&I = 0;
      %END;
      RUN;
```

Hyperslicing with SAS (Continued)

```
    %DO I = 0 %TO %EVAL(&LEVELS-1);

        PROC SGSCATTER DATA = TEMP;
        MATRIX &VAR/ GROUP = L&I DIAGONAL=(HISTOGRAM KERNEL);
        RUN;

    %END;

%MEND;
```

By creating many charts, we can then scroll through the charts to understand the changes. Below is the example generated using the macro with the CARS data using weight as the third dimension with 5 level.

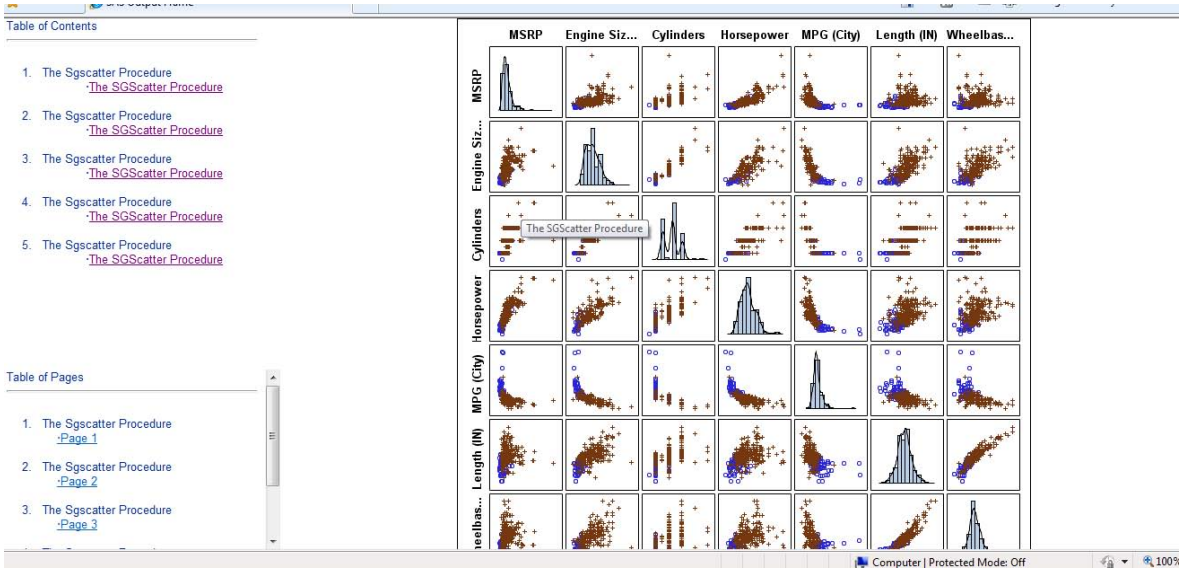

Figure 4: Level 1 of Hyperslice

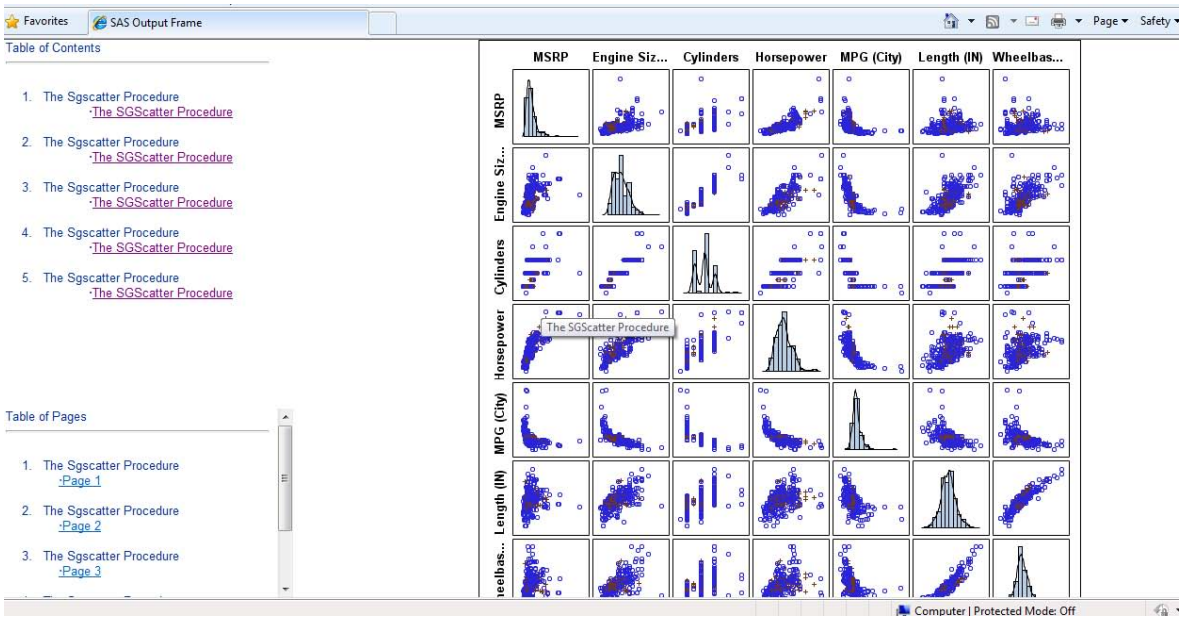

Figure 5: Level 4 of Hyperslice

You can see that with the highlighting coupled with the pages, one could slowly scroll and understand how the values changes through changing the third dimension. However, something the changing patterns is not sufficient to demonstrate the changes in the data.

APPROACH 2: REGRESSION LINE

Simple linear regressions between two variables produce excellent charts to demonstrate relations. In the case of hyperslice, we can see which parts of the data conform to the overall relations and which parts do not. Below is the adjusted macro to do regression lines.

```
/*HYPERSLICE*/

%MACRO HYPERSLICE(DATA,VAR,GROUP,LEVELS);

      PROC SORT DATA = &DATA;BY &GROUP;RUN;

      PROC RANK DATA = &DATA(KEEP = &VAR &GROUP) OUT = TEMP GROUPS = &LEVELS;
      VAR &GROUP;RANKS &GROUP.R;
      RUN;

      DATA TEMP;
      SET TEMP;
      %DO I = 0 %TO %EVAL(&LEVELS-1);
            IF &GROUP.R = &I THEN L&I = 1;ELSE L&I = 0;
      %END;
      RUN;

      %DO I = 0 %TO %EVAL(&LEVELS-1);

            PROC SGSCATTER DATA = TEMP;
            COMPARE X = (&VAR) Y = (&VAR) / GROUP = L&I REG;
            RUN;

      %END;

%MEND;
```

Below are the charts produced. The blues lines demonstrate the overall trend that is supplemented by the brown lines indicating the trends within the points being highlighted by the third dimension.

Hyperslicing with SAS (Continued)

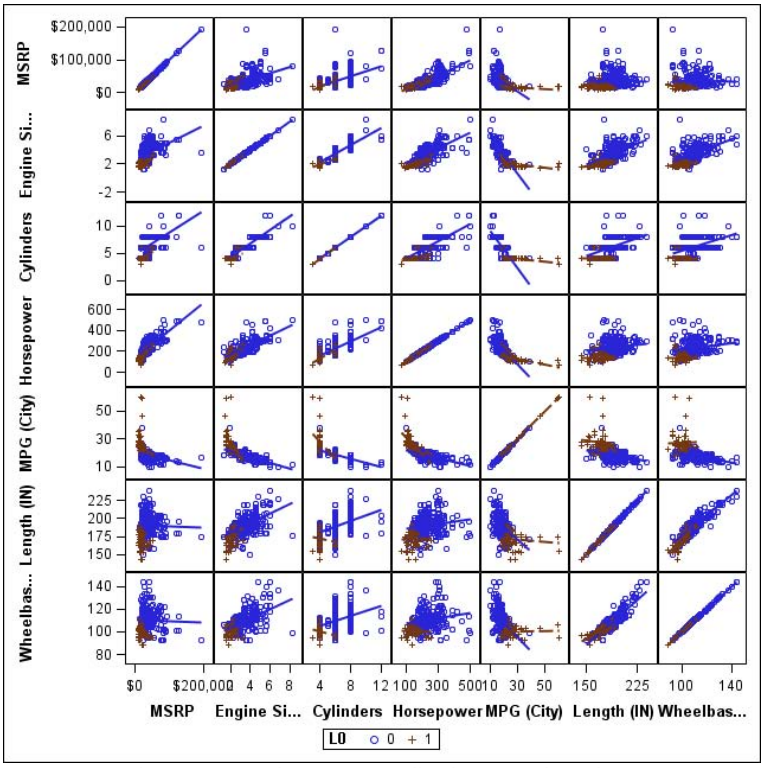

Figure 6: Level 1 of Hyperslice/Regression Lines

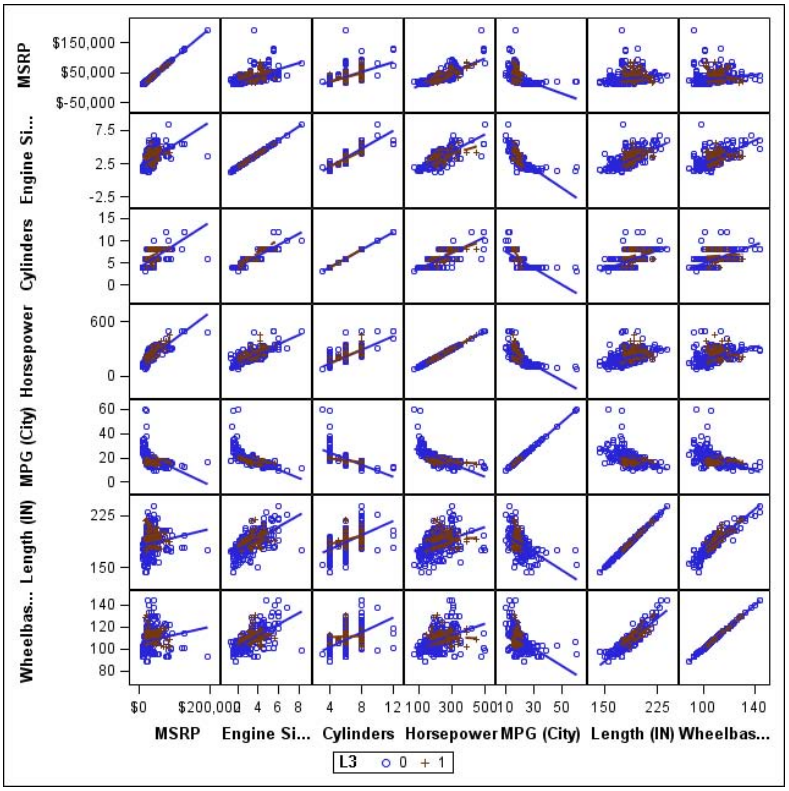

Figure 7: Level 4 of Hyperslice/Regression Lines

Hyperslicing with SAS (Continued)

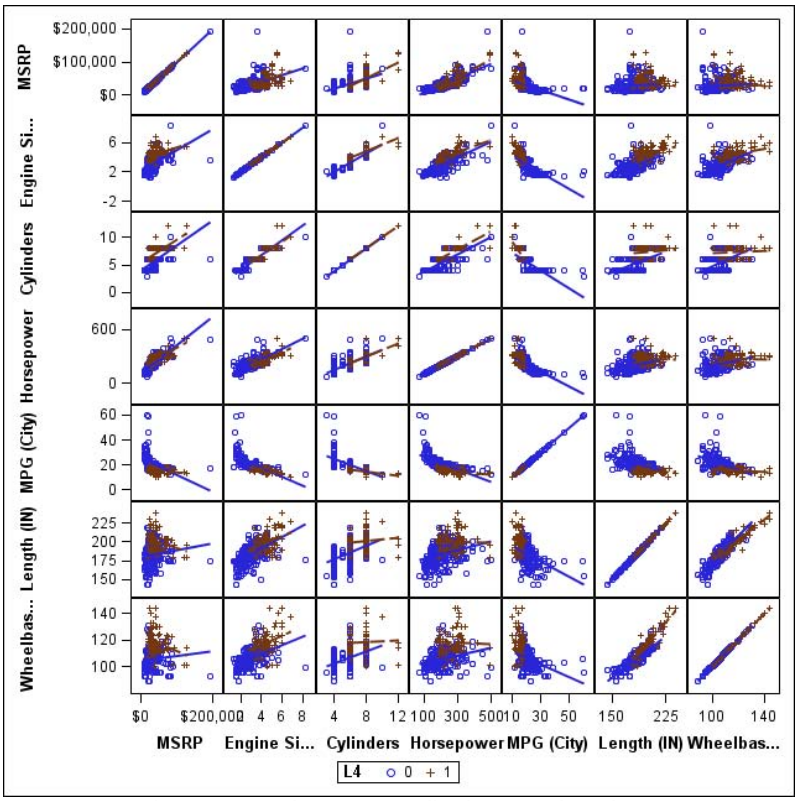

Figure 8: Level 5 of Hyperslice/Regression Lines

APPROACH 3: AF INTERACTIVE FRAME

However, while interactivity is extremely limited in terms of base SAS, additional level of interaction can be achieved by using SAS AF frames. While the technology has been deprecated, it nevertheless provides user with additional capabilities to select variables and the ease of changing variables for comparison purposes.

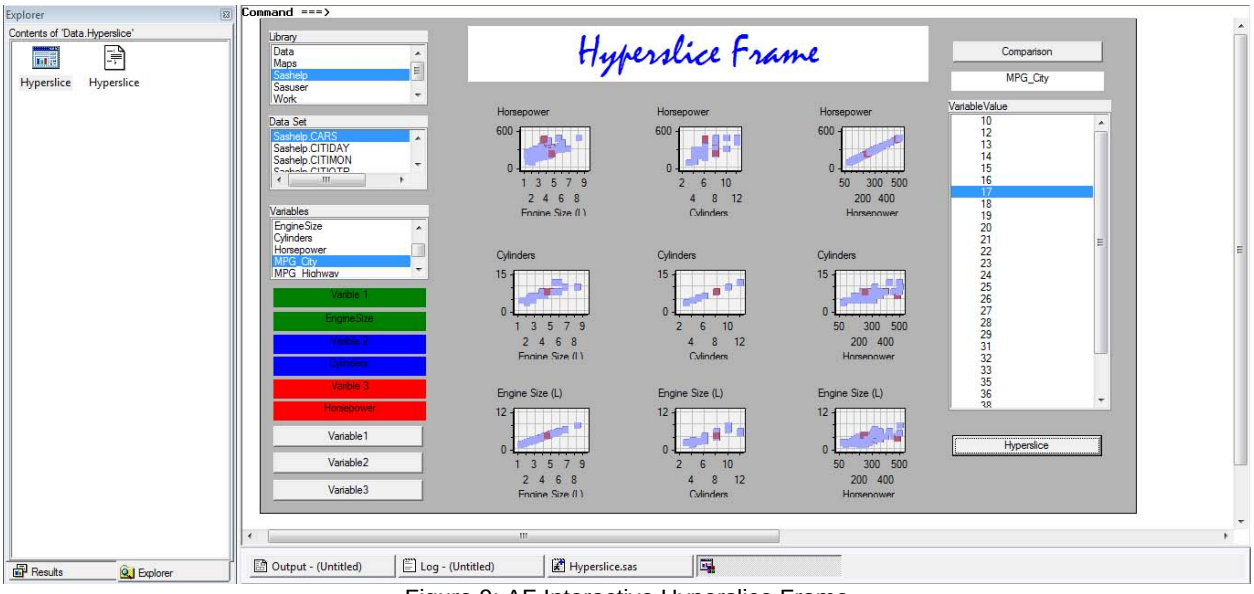

Figure 9: AF Interactive Hyperslice Frame

Hyperslicing with SAS (Continued)

JMP IMPLEMENTATION

Besides BASE SAS implementations, we explored the use of the JMP platform. JMP provides a very powerful visualization platform on which we can build on with the JMP Scripting Language (JSL). Starting from a basic scatter plot matrix, we added direct interaction through the Data Filter, giving us a great amount of flexibility in our interactions. Using the previous CARS example, we fix the MPG dimension as our filter value.

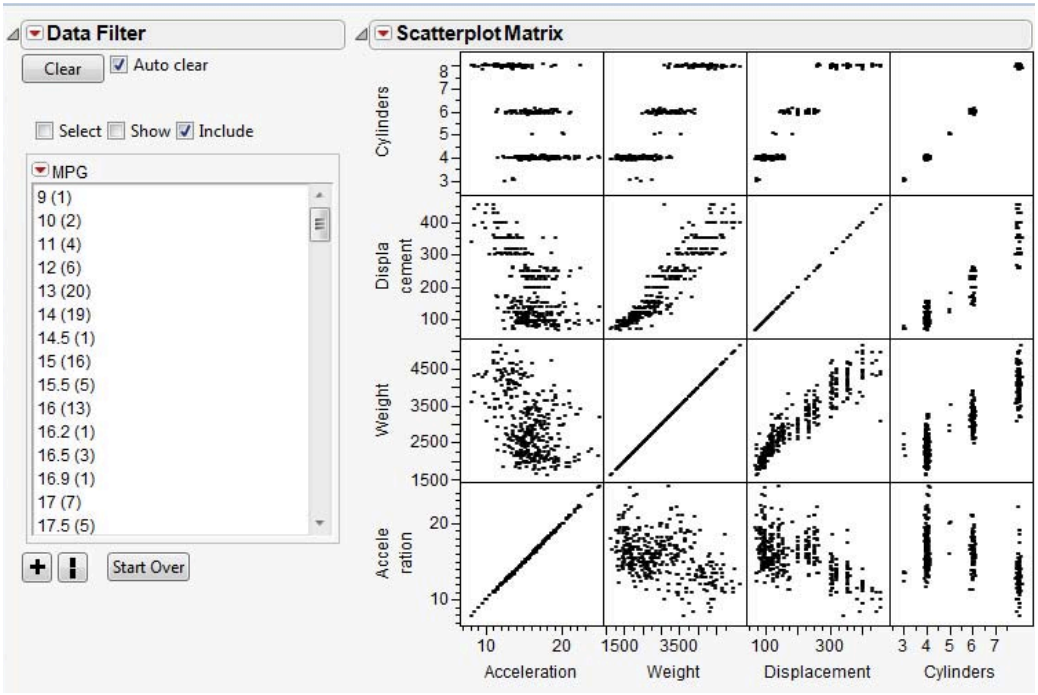

Figure 10: Initial JMP Hyperslice Frame

The Data Filter provides 3 levels of selection that can be combined for our exploration. By using the “Select” option, we can highlight the desired slice and observe them in comparison to the other points.

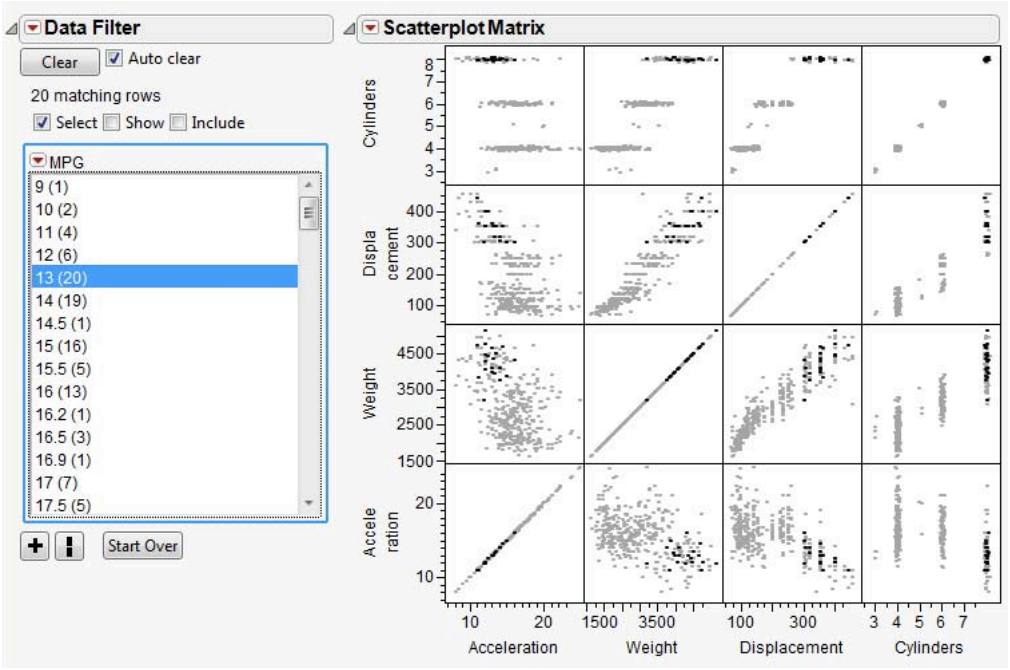

Figure 11: The “Select” Option

Hyperslicing with SAS (Continued)

By choosing the “Show” and “Include” options, we can specifically observe the slice and explore its dimensions clearer.

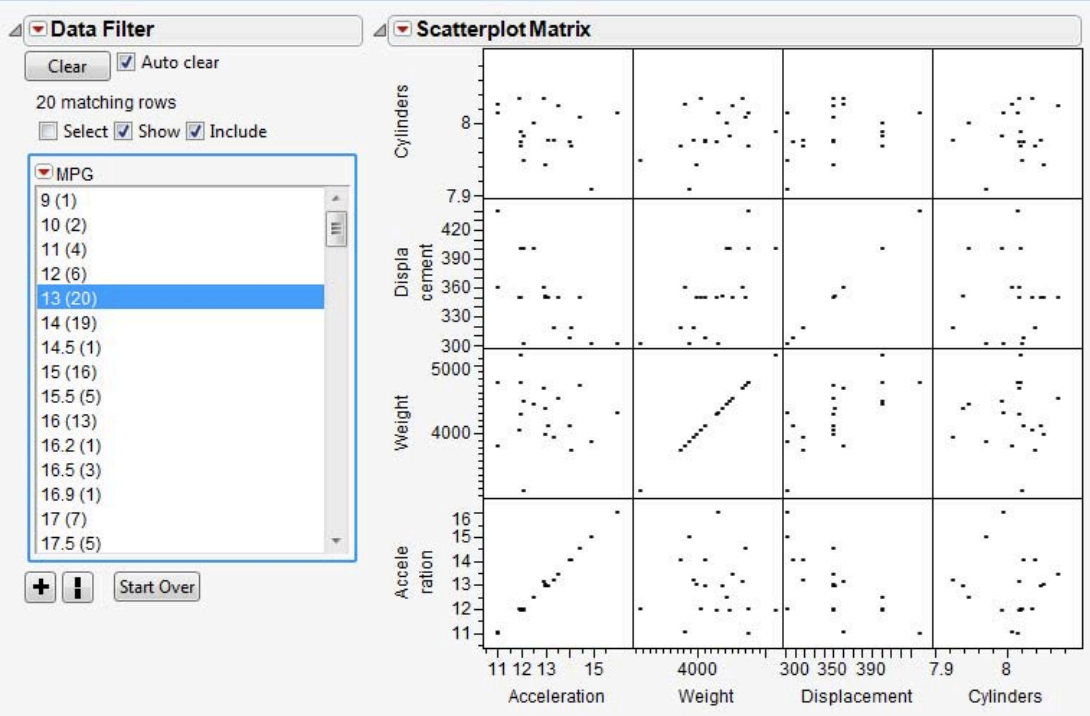

Figure 12: The “Show” and “Include” Options

If we choose to further explore this slice, we can use the Data Filter’s option to make this data a subset. From here, we can further focus our exploration through further use of Hyperslice or other techniques.

Another advantage of the Data Filter is the ability to animate through the different slices of the data. Using this with the “Select” option allows us to see how the changing slices progress compared to others. Using this with subsets further allows us to simulate the “dragging” interaction of the original Hyperslice.

Hyperslicing with SAS (Continued)

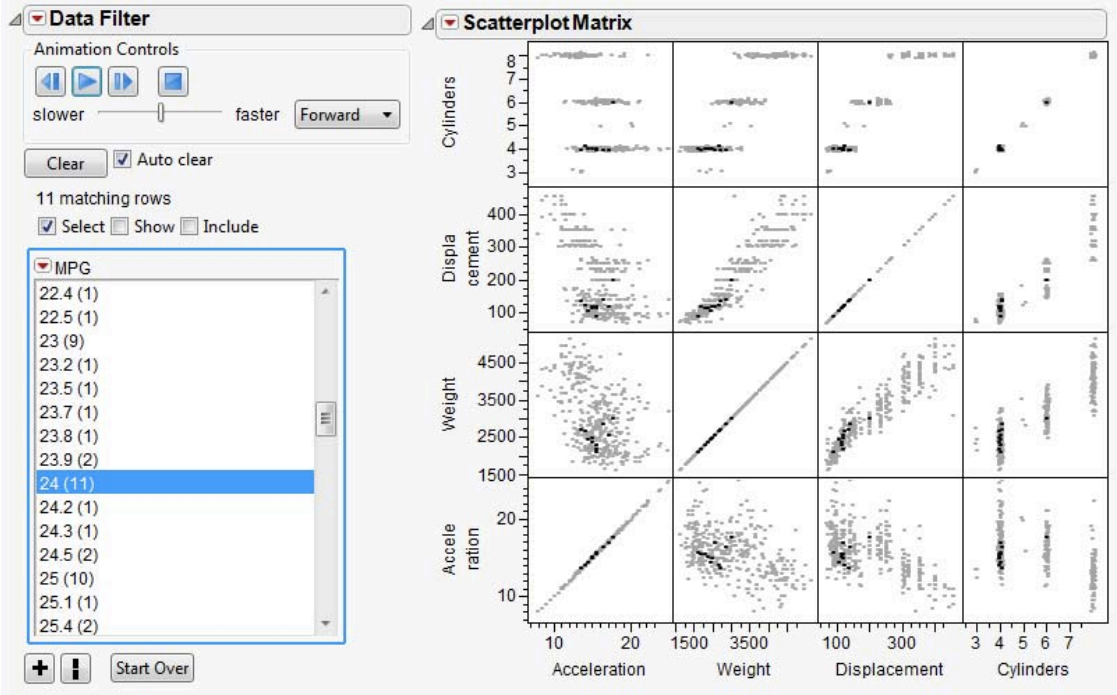

Figure 13: Animated Filter Progression with “Select” Option

CONCLUSION

Hyperslice is an effective technique to explore high dimensional data and thus, through this paper, we have shown multiple possible implementations of Hyperslice in SAS, further strengthening the ability to visualize and explore high-dimensional data effectively.

REFERENCES

[1] J. J. van Wijk and R.. D. van Liere, “Hyperslice,” in Proc. Visualization ’93, San Jose, CA, 1993, pp. 119–125.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Jonathon Khoo Kay Chin
Singapore Management University
School of Information System
80 Stamford Road
Singapore 179802
kckhoo.2008@smu.edu.sg

Koo Ping Shung
Singapore Management University
School of Information System
80 Stamford Road
Singapore 179802
pskoo@smu.edu.sg