

Enhancing Clinical Data Analysis: The PROC GREPLAY Advantage

Vaisint Singh Muthudurai, Zifo, Chennai, India

Kalyane Subramanian, Zifo, Chennai, India

ABSTRACT

In clinical trials, summarizing extensive data for each subject often results in dense, hard-to-review tables. Graphical representations offer clearer insights by displaying each subject's response separately, but individual graphs can lead to lengthy reports. PROC GREPLAY in SAS addresses this by customizing graph layouts, allowing multiple graphs on a single page. This customization enhances the accessibility of findings, making them more effective for scientific publications, conference presentations, and stakeholder communications.

This method conserves space and improves the ability to compare responses between subjects or track changes within the same subject over time. By leveraging PROC GREPLAY, we can enhance data visualization and facilitate more efficient and effective analysis of clinical trial results. In this paper, we will discuss the application of PROC GREPLAY in SAS in detail, providing practical insights, challenges and best practices.

INTRODUCTION

PROC GREPLAY in SAS is a procedure used to combine and display multiple graphs in a single layout, such as a single page within a PDF document. It's primarily used to enhance the presentation of graphs by arranging them logically, resizing, and potentially adding borders, titles, or footnotes. It acts as a tool to organize and present graphics generated by other SAS/GRAPH procedures, like GPLOT or GCHART, or even newer procedures like SGPLOT. The core function of PROC GREPLAY is to combine multiple graphs into one output, allowing for side-by-side comparisons or the creation of more complex visualizations. It allows for precise control over the arrangement of graphs on the page, including positioning, sizing, and scaling. PROC GREPLAY can be used to create output in various formats, including PDF, RTF and image formats such as PNG and JPEG, which are particularly useful for creating reports and presentations. It integrates seamlessly with the Output Delivery System (ODS), allowing for easy output to different formats and destinations. It essentially replays previously created graphics from a catalog, making it efficient for reusing and combining existing visualizations. PROC GREPLAY can work with templates that define the layout of the output, allowing for consistent and reusable graph arrangements.

MULTIPLE OF GRAPHS INTO A SINGLE OUTPUT

We can display multiple graphs on a single page using predefined or custom templates (called "GREPLAY templates"). This is especially useful for creating dashboards or comparative visualizations.

Example 1

Assume we want to display 4 graphs on the same page. Firstly, we tell PROC GREPLAY to create a template with five panels. The four panels are small and equal in size. The fifth panel is a larger, full-size panel that can be used to display a common title or footnote for the entire template.

Step 1: Create a Template

```
options reset=all border;

proc greplay tc=work.tempcat nofs;
tdef newtemp des="Five panel template"

/*Panel 1: Lower Left Quadrant*/
1 / llx = 0    lly = 10
   ulx = 0    uly = 50
   urx = 50   ury = 50
   lrx = 50   lry = 10
   color =    navy

/*Panel 2: Upper Left Quadrant*/
2 / llx = 0    lly = 50
   ulx = 0    uly = 90
```

```

    urx = 50    ury = 90
    lrx = 50    lry = 50
    color = lime

/*Panel 3: Upper Right Quadrant*/
3 / llx = 50    lly = 50
    ulx = 50    uly = 90
    urx = 100   ury = 90
    lrx = 100   lry = 50
    color = yellow

/*Panel 4: Lower Right Quadrant*/
4 / llx = 50    lly = 10
    ulx = 50    uly = 50
    urx = 100   ury = 50
    lrx = 100   lry = 10
    color = cyan

/*Panel 5: Container for Title and Panels 1-4*/
5/ llx = 0    lly = 0
    ulx = 0    uly = 100
    urx = 100  ury = 100
    lrx = 100  lry = 0
    color = black

    template newtemp;

    list template;
quit;

```

The `options reset=all border;` statement resets all previously defined graphics options and adds a border around the output display. The `proc greplay` procedure is then initiated with the `tc=work.tempcat` option, which specifies the temporary template catalog where the layout will be stored. The `des="Five panel template"` option provides a descriptive label for documentation purposes. Within this procedure, the `TDEF` statement is used to define a new template named `NEWTEMP`, which is saved in the specified catalog. Each panel in the template is defined by its number and the coordinates of its four corners: lower left (`llx`, `lly`), upper left (`ulx`, `uly`), upper right (`urx`, `ury`), and lower right (`lrx`, `lry`). These coordinates determine the position and shape of each panel within the layout. Additionally, the `COLOR=` option can be used to draw a border around each panel in the specified color, enhancing the visual structure of the output.

Step 2: Generate Individual Graphs

```

data set.cdata (drop=name);
length Gender $ 6;
set sashelp.class;

if not missing(SEX) and SEX eq "F" then Gender="Female";
else if not missing(SEX) and SEX eq "M" then Gender = "Male";
else Gender = SEX;

run;

proc sort data= set.cdata out = set.cdatal;
by weight height;
run;

goptions reset=all hsize=2.75in vsize=2.06in;

```

```

axis1 label=none style=0 major=none value=none;
axis2 label=("Age");
axis3 label=("Height") order=50 to 75 by 5;
axis4 label=("Weight") order=50 to 150 by 25 minor=(n=1);

legend1 label=none value=("Male" "Female") Position = (right middle
    outside) across = 1;

legend2 label = none value = ("Male" "Female");

symbol1 i=join;

proc gchart data=set.cdata1 gout=set.exdat;

    vbar age/discrete hminor=0 subgroup=gender
        inside=freq raxis=axis1 maxis=axis2
        noframe legend=legend1;

    run;

    hbar age/ discrete sumvar=height mean
        meanlabel="Avg.Height" vminor=0
        raxis=axis1 maxis=axis2;

    run;

    pie gender/ noheading legend=legend1 percent=inside;

    run;

    proc gplot data=class gout=set.excat;
        plot height*weight=gender/ vminor=1 vaxis=axis3
            haxis=axis4 legend=legend2;

    run;

quit;

```

The minor=(n=1) option specifies that one minor tick mark should be drawn between each pair of major tick marks on the axis, enhancing the granularity of the axis scale. The i=join option instructs SAS to connect the data points with straight lines, resulting in a continuous line graph. This is particularly useful for visualizing trends over time or across ordered categories, as it provides a clear and smooth representation of data progression.

Step 3: Create a Title Slide (Optional)

```

proc gslide gout = set.excat;
    title1 h=2 justify=center "Combined Graph Analysis" ;
    footnote1 h=1 justify=left "Graphs produced using PROC GREPLAY";
run;

```

PROC GSLIDE is used to add a common title or footnote that spans the entire page.

Step 4: Replay Graphs into the Template

```

proc greplay gout = new igout=set.excat nofs
    tc=tempcat template = newtemp;
    treplay 1:gchart 2:gchart1 3:gchart2 4:gplot 5: gslide;

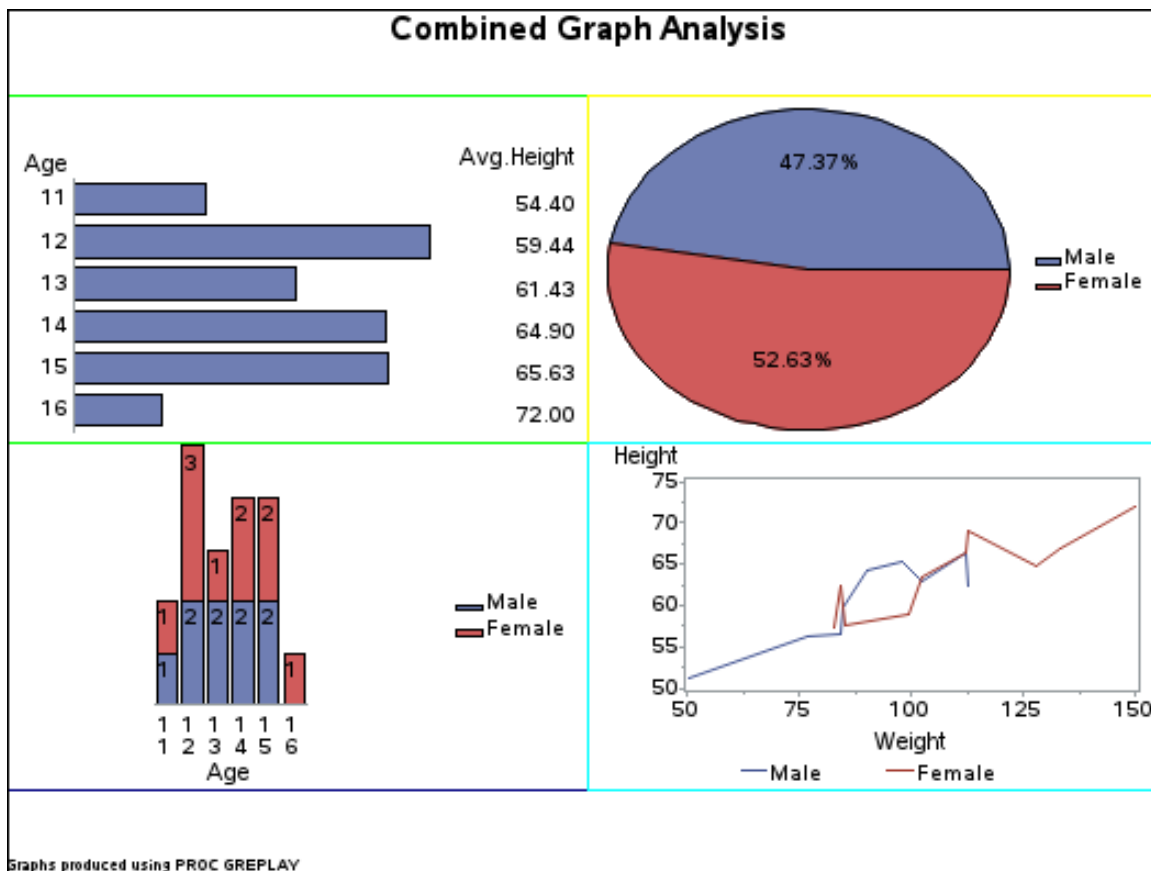
quit;

```

In the PROC GREPLAY step, the TREPLAY statement is used to assign specific graphics output to each panel in the template you defined earlier. For example, 1:gchart assigns the graph named gchart to panel 1, 2:gchart1 assigns gchart1 to panel 2, and so on. The names like gchart1, gchart2, gplot, and gslide refer to the names of graphics output that were previously created and stored in the graphics catalog. By specifying these names, you control which graph appears in each panel of your custom layout. This allows you to display multiple related graphs together in a single,

organized output. the NOFS option stands for "No Full Screen". It is used to disable the interactive full-screen GREPLAY interface, allowing the procedure to run in batch mode.

Output:



Example 2

By using PROC GREPLAY, we can consolidate multiple subject-specific graphs into a single output layout. The following example is created using dummy sample data to demonstrate the functionality.

Step 1: Create a Template

```
options reset=all border;

proc greplay tc=tempcat nofs;
  tdef newtemp des="7 panel template"

  1 / llx = 0      lly = 51
      ulx = 0      uly = 90
      urx = 33     ury = 90
      lrx = 33     lry = 51

  2 / llx = 34     lly = 51
      ulx = 34     uly = 90
      urx = 66     ury = 90
      lrx = 66     lry = 51

  3 / llx = 67     lly = 51
      ulx = 67     uly = 90
```

```

        urx = 100    ury = 90
        lrx = 100    lry = 51

4 / llx = 0        lly = 10
    ulx = 0        uly = 50
    urx = 33       ury = 50
    lrx = 33       lry = 10

5 / llx = 34       lly = 10
    ulx = 34       uly = 50
    urx = 66       ury = 50
    lrx = 66       lry = 10

6 / llx = 67       lly = 10
    ulx = 67       uly = 50
    urx = 100      ury = 50
    lrx = 100      lry = 10

7 / llx = 0        lly = 0
    ulx = 0        uly = 100
    urx = 100      ury = 100
    lrx = 100      lry = 0;

quit;

proc sql noprint;
    select sum(ceil(sum(min(score),-10)), -mod(ceil(sum(min(score),-
        10)),10)) into: min1 from datascore1;
    select ceil(sum(max(score),10)) into: max1 from datascore1;
quit;

```

This PROC SQL block is used to calculate dynamic axis limits for plotting subject scores.

Step 2: Generate Individual Graphs

```

ods pdf file = "&path.\points.pdf";

goptions reset=all hsize=3.5in vsize=2.5in;
axis1 label=none order=&min1. to &max1. by 10;
axis2 label=none order=0 to 48 by 12;
symbol1 i=join value=dot;

proc gplot data=datascore1;
    plot score*visitnum /frame nolegend vminor=1;
    by subjectid;
run;

```

The symbol1 option defines the plotting style by connecting data points with lines (i=join) and representing each point with a dot (value=dot). The vminor=1 option specifies that one minor tick mark should be displayed between each major tick mark on the Y-axis, enhancing the granularity of the axis. The frame option adds a border around the plot area for better visual clarity, while the nolegend option suppresses the legend in the plot to keep the output clean. Additionally, the ods pdf statement activates the PDF output destination, and the file= option specifies the full path and filename where the PDF output will be saved.

Step 3: Create a Title Slide (Optional)

```

proc gslide gout=GSEG;
    title1 h=2 "subjects score upto week 4";
    footnotel h=1 "Week";

```

```
run;
```

Step 4: Replay Graphs into the Template

```
%macro greplay;
  proc sql noprint;
    select count(distinct SUBJECTID) into :npats from Datascore1;
  quit;
  %do i = 1 %to &npats %by 6;
    %if %eval(&i) eq 1 %then %let j = 1;
    %else %let j = %eval((&i / 6)+1);

    proc greplay igout=gseg gout= new tc=tempcat nofs;
      template newtemp;

      %if %eval(&i+5) le &npats %then %do;
        treplay
          %if %eval(&i) eq 1 %then 1:gplot;
          %else 1:gplot%eval(&i-1);
          2:gplot%eval(&i)
          3:gplot%eval(&i+1)
          4:gplot%eval(&i+2)
          5:gplot%eval(&i+3)
          6:gplot%eval(&i+4)
          7:gslide
        ;
      %end;

      %else %do;
        treplay
          %if %eval(&i) le &npats %then 1:gplot%eval(&i-1)
          %if %eval(&i+1) le &npats %then 2:gplot%eval(&i)
          %if %eval(&i+2) le &npats %then 3:gplot%eval(&i+1)
          %if %eval(&i+3) le &npats %then 4:gplot%eval(&i+2)
          %if %eval(&i+4) le &npats %then 5:gplot%eval(&i+3)
          %if %eval(&i+5) le &npats %then 6:gplot%eval(&i+4)
          7:gslide
        ;
      %end;
    ;
  quit;
%end;

%mend greplay;

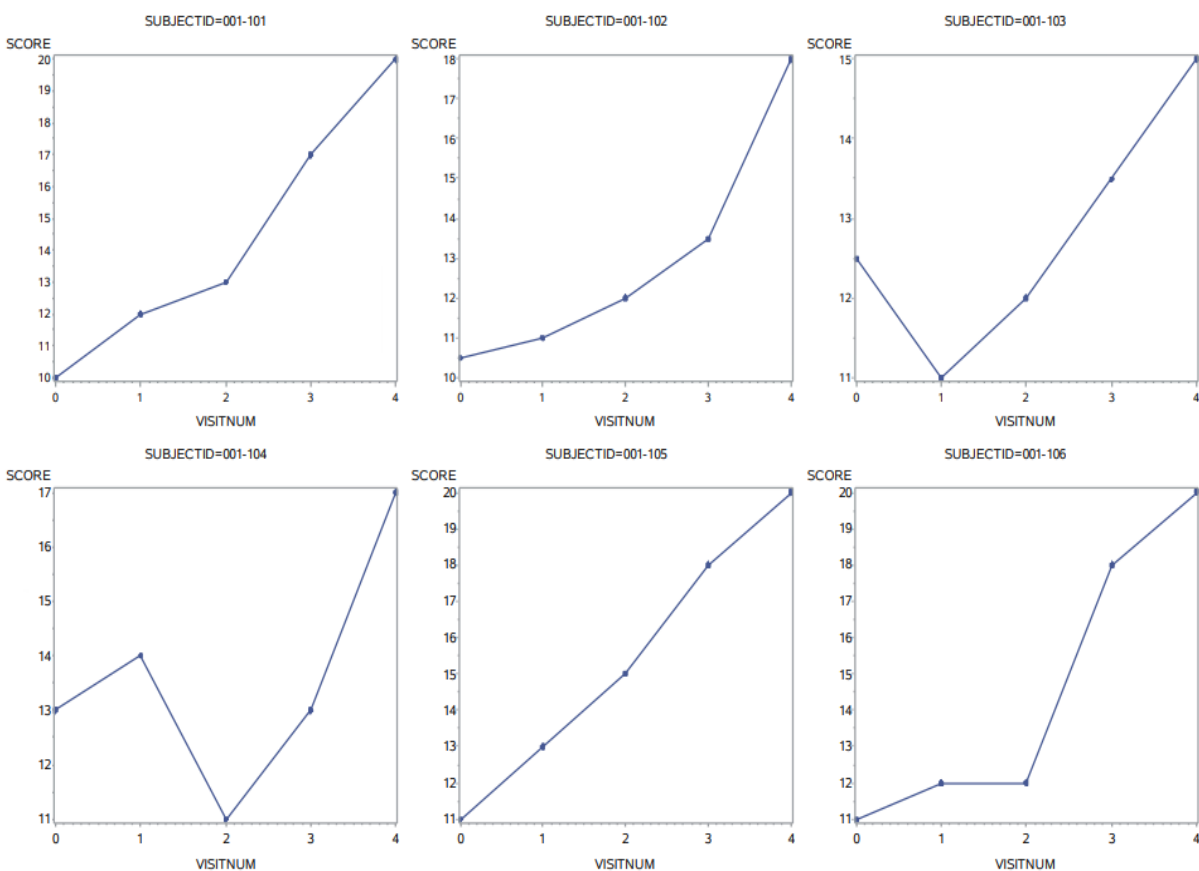
%greplay;

ods pdf close;
```

This step is designed to efficiently organize and display multiple subject-specific graphs within a single output layout. After generating individual plots and defining a custom 6-panel template, this macro automates the process of replaying the graphs into the appropriate panels. It begins by counting the number of unique subjects and then loops through them in batches of six. For each batch, it uses PROC GREPLAY to assign the corresponding plots to panels 1 through 6 of the templates, while panel 7 is reserved for a shared title slide created using PROC GSLIDE. The macro also includes logic to handle partial batches when fewer than six subjects remain, ensuring that only available plots are replayed. This structured approach allows for clean, comparative visualizations across subjects and is especially useful for summarizing clinical or longitudinal data in a visually coherent format.

Output:

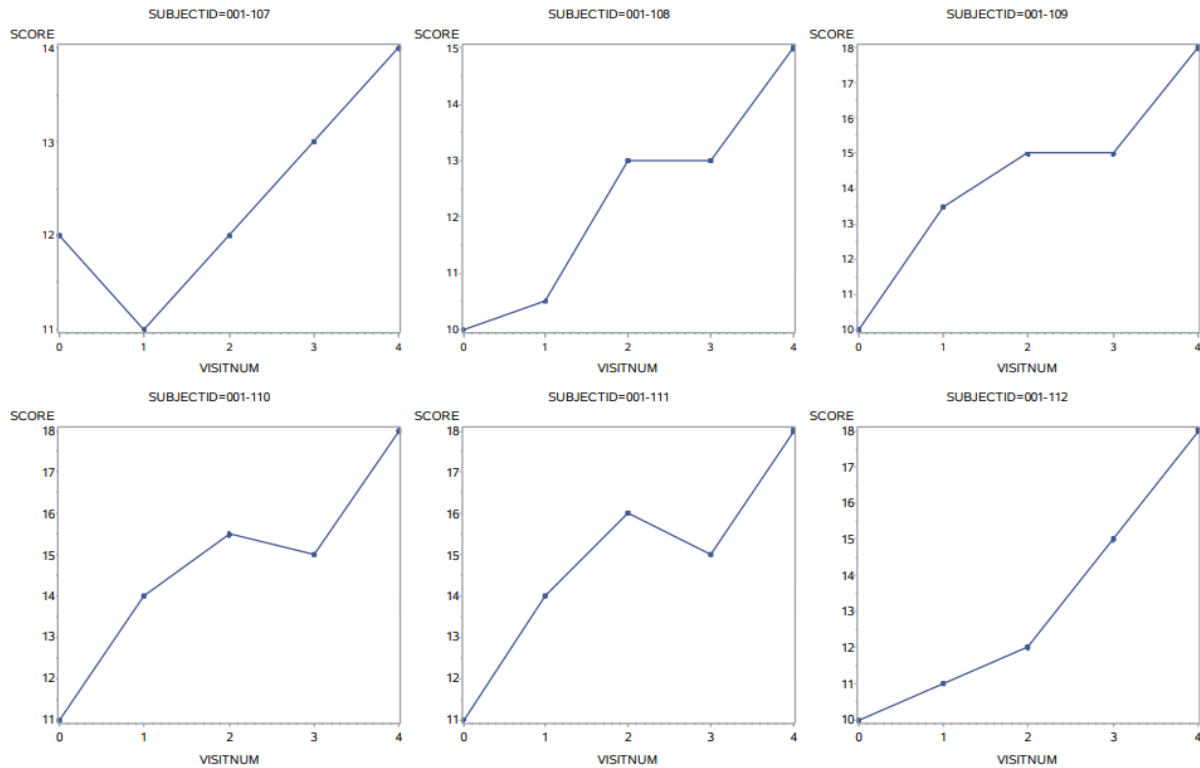
subjects score upto week 4



Week

Page 1

subjects score upto week 4



Week

Page 2

By using PROC GREPLAY, we can consolidate multiple subject-specific graphs into a single output layout. This approach allows for efficient analysis by presenting individual responses side by side, reducing report length while enhancing clarity. This approach provides a clean, organized way to present multiple subject-level visualizations in a single output, making it easier to compare and interpret data across subjects.

KEY FEATURES OF PROC GREPLAY:

- **Template-based Layouts:** You can use predefined or custom templates (like V2, H2, T2R2, etc.) to arrange multiple plots on a single page.
- **Catalog Management:** It works with graphics catalogs (usually created by PROC GPLOT, GCHART, etc.), allowing you to replay and rearrange existing plots without regenerating them.
- **Integration with PROC GSLIDE:** This is especially useful for adding custom slides with titles, annotations, or footnotes between plots—ideal for presentations or reports.
- **Overlaying and Replaying:** You can replay plots in different orders or overlay them for comparison, which is helpful in longitudinal studies or clinical trial reporting.

ADVANTAGES

PROC GREPLAY in SAS is a versatile tool that enhances the presentation and analysis of graphical outputs by allowing multiple plots to be arranged on a single page. This is especially useful when working with subject-wise or time-series data, where individual graphs can become overwhelming and lead to lengthy reports. By using custom templates,

PROC GREPLAY enables users to organize and replay graphics stored in catalogs, making it easier to compare and interpret results across subjects or conditions. It also supports integration with slides created using PROC GSLIDE, allowing for the inclusion of titles, annotations, and footnotes to create professional, publication-ready reports. Overall, PROC GREPLAY streamlines the review process, improves clarity, and adds flexibility to graphical reporting in SAS.

LIMITATIONS

While PROC GREPLAY offers valuable functionality for organizing and displaying multiple graphics in SAS, it does come with several limitations. One of the main challenges is the complexity involved in creating custom templates, which require manual specification of coordinates and can be time-consuming to design and adjust. The procedure also relies heavily on graphics catalogs, which can be difficult to manage, especially when dealing with large volumes of output. Additionally, the output generated by PROC GREPLAY is static and lacks interactivity, limiting its usefulness for exploratory data analysis. Styling and customization options are relatively basic compared to modern visualization tools, and it may not integrate well with newer SAS graphics procedures that use the ODS Graphics system. For new users, the learning curve can be steep due to the need to understand SAS/GRAPH concepts like templates, panels, and catalog management. Overall, while PROC GREPLAY is effective for structured graphical reporting, it may not be the best choice for dynamic or highly customized visualizations.

CONCLUSION

In clinical data analysis, presenting subject-wise graphical outputs in a clear and concise manner is essential for effective interpretation and decision-making. PROC GREPLAY in SAS significantly enhances this process by allowing multiple individual graphs to be organized into a single, structured layout. This not only reduces the length and complexity of reports but also facilitates easier comparison across subjects or treatment groups. By leveraging custom templates and integrating slides for annotations, PROC GREPLAY transforms dense graphical data into accessible visual summaries, making it a valuable tool for clinical research and reporting.

REFERENCES

SAS Support: SAS/GRAPH(R) 9.2: Reference, Second edition.

CONTACT INFORMATION

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

1st Author Name: Vaisint Singh Muthudurai

1st Author Email: Vaisint.M@zifornd.com

2nd Author Name: Kalyane Subramanian

2nd Author Email: Kalyane.S@zifornd.com

Company: Zifo

Address: 6th Floor, Block 2, DLF SEZ IT Park, Porur, Chennai, Tamil Nadu 600089

City / Postcode: Chennai / 600089

Work Phone: +91 44 43114002

Website: <https://zifornd.com/>

Brand and product names are trademarks of their respective companies.