PharmaSUG China 2023 - Paper DV-124

Sunburst Plot Analysis and Submission to FDA with Python

Wei (Tony) Zhang, Pfizer (China) Research and Development Co.,Ltd.

ABSTRACT

The sunburst plot is ideal for displaying hierarchical data. It visualizes hierarchical data spanning outwards radially from root to leaves by levels. Each level of the hierarchy is represented by one ring or circle and all rings show how the outer rings relate to the inner rings through hierarchy. It is very efficient to create sunburst plot in Python using interactive data visualization package "Plotly". The plot output and Python code to generate the sunburst plot analysis were also submitted to FDA in a real case. In this article, we present how to generate and customize the sunburst plot with Python Potly package.

INTRODUCTION

Sunburst plot is a fantastic way to visualize and understand hierarchical or sequential data. The main use cases of the sunburst plot analysis are: (1). Understand largest contributors to a whole and the proportions of the components. (2). Understand flow of individuals among sequential steps of a process.

The following Figure 1 shows the example of the sunburst plot which visualizes the hierarchical data.

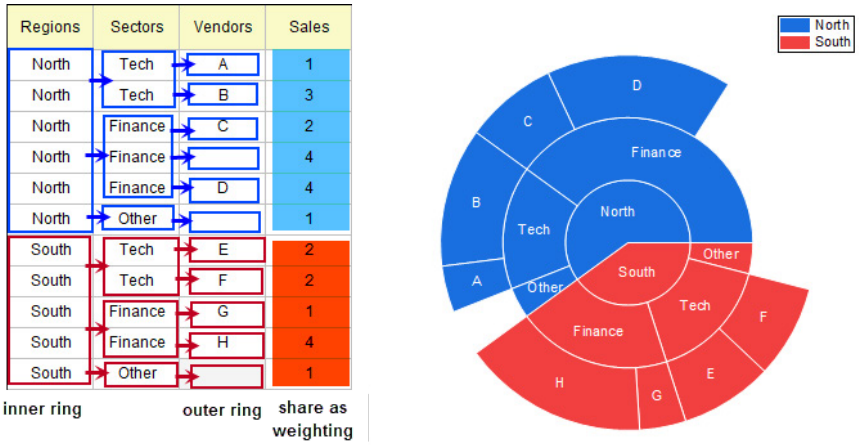

Figure 1: Visualization of Hierarchical Data with Sunburst Plot

The sunburst plot is not ideal for time-series or continuous data but rather for data of hierarchical or sequential nature. When the sponsor submits the clinical trail data to the agency, the sunburst plot analysis may be requested to visualize the hierarchical or sequential nature of the data for safety or efficacy purpose.

Currently, SAS cannot have an easy way to generate the sunburst plot. R or Python is a good choice to generate the sunburst plot with Plotly package. In Python, the Plotly Python library is an interactive, open-source plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases. Plotly Express is the easy-to-use, high-level interface to Plotly, which operates on a variety of types of data and produces easy-to-style figures.

PLOTLY OPEN SOURCE GRAPHING LIBRARY FOR PYTHON

Plotly's Python graphing library makes interactive, publication-quality graphs. Examples of how to make line plots, scatter plots, area charts, bar charts, error bars, box plots, histograms, heatmaps, subplots, multiple-axes, polar charts, and bubble charts. Plotly.py is free and open source and you can view the source, report issues or contribute on GitHub.

INSTALLATION

plotly may be installed using pip:

$ pip install plotly

or conda:

$ conda install -c plotly plotly

This package contains everything you need to write figures to standalone HTML files.

BASIC SUNBURST PLOT WITH PLOTLY.EXPRESS

Plotly Express is the easy-to-use, high-level interface to Plotly, which operates on a variety of types of data and produces easy-to-style figures. With px.sunburst, each row of the DataFrame is represented as a sector of the sunburst.

```
import plotly.express as px
data = dict(
   character=["Eve", "Cain", "Seth", "Enos", "Noam", "Abel", "Awan", "Enoch", "Azura"],
   parent=["", "Eve", "Eve", "Seth", "Seth", "Eve", "Eve", "Awan", "Eve" ],
   value=[10, 14, 12, 10, 2, 6, 6, 4, 4])
fig = px.sunburst(data, names='character',
                        parents='parent', values='value'
                        )
fig.show()
```

Then figure 2 is the output of the simple sunburst plot by the above Python code.

Figure 2: A simple Sunburst Plot

SUNBURST OF A RECTANGULAR DATAFRAME WITH PLOTLY.EXPRESS

Hierarchical data are often stored as a rectangular dataframe, with different columns corresponding to different levels of the hierarchy. px.sunburst can take a path parameter corresponding to a list of columns. Note that id and parent should not be provided if path is given.

```
import plotly.express as px

df = px.data.tips()

fig = px.sunburst(df, path=['day', 'time', 'sex'], values='total_bill')

fig.show()
```

Then figure 3 is the output of the plot by the above Python code.

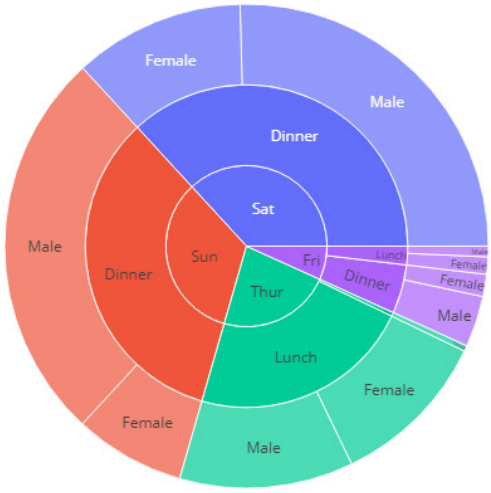

Figure 3: Sunburst Plot for Rectangular Dataframe

DATA STRUCTURE FOR SUNBURST PLOT

Hierarchical data should be structured by columns. This means each level in the sunburst plot should be stored as column in the dataframe. This is the example of a dummy dataset with correct data structure.

Table View

	USUBJID	SU1	SU2	FD1	CENTER
1	1	Grade 1 without Toci	No CRS	No Dose	Enrolled
2	2	No CRS	No CRS	No CRS	Enrolled
3	3	Grade 1 with Toci	Grade 1 with Toci	Grade 1 without Toci	Enrolled
4	4	No CRS	Grade 1 without Toci	No CRS	Enrolled
5	5	Grade 1 with Toci	No CRS	Grade 2 with Toci	Enrolled
6	6	No CRS	Grade 1 without Toci	No CRS	Enrolled
7	7	No CRS	Grade 1 without Toci	Grade 2 with Toci	Enrolled
8	8	No CRS	No CRS	No CRS	Enrolled
9	9	Grade 1 without Toci	Grade 1 without Toci	No CRS	Enrolled
10	10	No CRS	No CRS	No CRS	Enrolled
11	11	Grade 3 with Toci	No CRS	No CRS	Enrolled
12	12	No CRS	No CRS	Grade 1 with Toci	Enrolled
13	13	No CRS	Grade 1 with Toci	No CRS	Enrolled
14	14	Grade 1 with Toci	No CRS	No CRS	Enrolled
15	15	Grade 2 with Toci	Grade 1 with Toci	No CRS	Enrolled
16	16	Grade 1 with Toci	No CRS	No CRS	Enrolled
17	17	Grade 1 without Toci	No CRS	No CRS	Enrolled
18	18	No CRS	Grade 1 without Toci	No CRS	Enrolled
19	19	No CRS	No CRS	No CRS	Enrolled
20	20	No CRS	No CRS	No CRS	Enrolled
21	21	Grade 1 without Toci	No CRS	No CRS	Enrolled
22	22	No CRS	Grade 1 without Toci	No CRS	Enrolled
23	23	Grade 2 without Toci	No CRS	No CRS	Enrolled
24	24	No CRS	Grade 1 with Toci	No CRS	Enrolled
25	25	Grade 1 with Toci	Grade 1 with Toci	No Dose	Enrolled
26	26	Grade 1 with Toci	No Dose	No Dose	Enrolled
27	27	Grade 2 with Toci	No CRS	No CRS	Enrolled
28	28	Grade 2 with Toci	No CRS	No Dose	Enrolled
29	29	No CRS	No CRS	No Dose	Enrolled
30	30	Grade 1 with Toci	Grade 1 without Toci	No Dose	Enrolled
31	31	Grade 1 without Toci	No Dose	No Dose	Enrolled
32	32	Grade 1 with Toci	No CRS	No CRS	Enrolled
33	33	Grade 2 with Toci	No CRS	Grade 1 without Toci	Enrolled
34	34	No CRS	No CRS	No CRS	Enrolled
35	35	No CRS	No Dose	No Dose	Enrolled
36	36	Grade 1 with Toci	Grade 1 without Toci	No CRS	Enrolled
37	37	No CRS	Grade 1 without Toci	No CRS	Enrolled

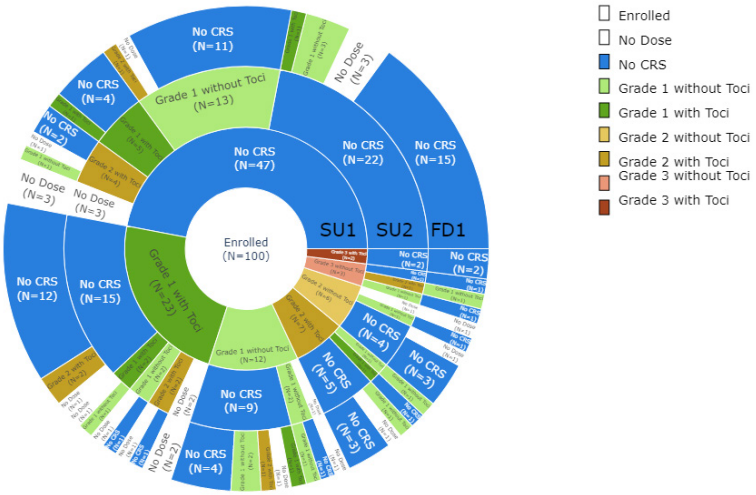

Figure 4: Dummy Dataset with Correct Data Structure for Sunburst Plot

In this dummy dataset, SU1, SU2, FD1 are three levels of sunburst plot from root to leaves as shown in Figure 4. The center roll is the enrolled information. This dummy set was created using below R coding.

```
library(SASxport)

library(tidyverse)

#dummy dataset creation using sunburst plot format

list<-c("No Dose", "No CRS","Grade 1 without Toci", "Grade 1 with Toci",  "Grade 2 without Toci" , "Grade 2 with To
ci", "Grade 3 without Toci" , "Grade 3 with Toci"  )

probs.su1<-c(0,0.6,0.15,0.1,0.05,0.05,0.025,0.025)

probs.su2<-c(0.05,0.65,0.2,0.05,0,0.05,0,0)

probs.fd1<-c(0.1,0.7,0.1,0.05,0.025,0.025,0,0)

set.seed(424242)

dat<-data.frame(USUBJID=seq(1:100),

                SU1=sample(x=list, prob=probs.su1,size=100,replace=TRUE),
```

```
        SU2=sample(x=list, prob=probs.su2,size=100,replace=TRUE),
        FD1 = sample(x=list, prob=probs.fd1,size=100,replace=TRUE),
        CENTER="Enrolled")%>%
 mutate(FD1=ifelse(SU2=="No Dose","No Dose",FD1))
setwd("C:/Users/PharmaSUG China 2023/dummy_sunburst")
write.xport(dat, file ="dummy_sunburst.xpt")
```

This dummy dataset is the example of CRS (Cytokine Release Syndrome) Analysis by Dose as shown below in Figure 5.

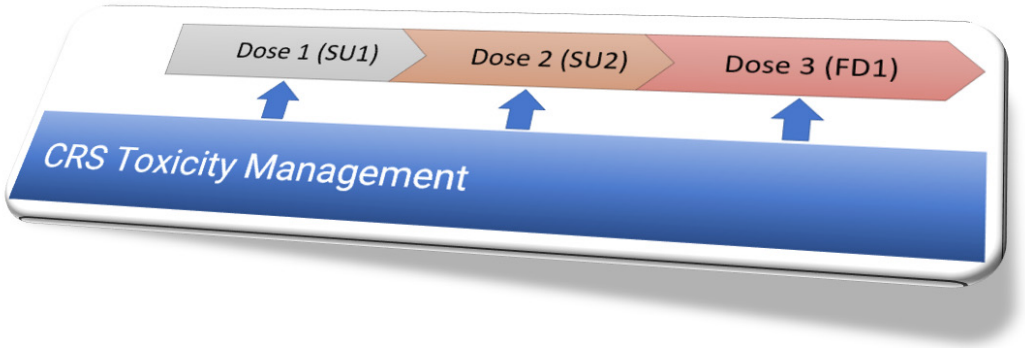

Figure 5: CRS Toxicity Analysis by Dose

MAIN BODY OF SUNBURST PLOT

Sunburst charts are fairly straightforward to build using Plotly Express, especially from a Pandas dataframe. The key things we have to specify are the dataframe using the data_frame argument, the columns to use and the hierarchy of them with the path argument and the column that will determine the size of the parts in our sunburst chart

The code of main body of sunburst plot is shown below, where 'SEQ' is assigned as 1 (SEQ=1) for all observations of the dataframe to calculate the total number of subjects in each level of the hierarchical data.

```
import plotly.express as px
import pandas as pd
df['SEQ']=1
fig = px.sunburst(df, path=['CENTER', 'SU1', 'SU2', 'FD1'], values='SEQ')
fig.show()
```

UPDATE FIGURE TRACE FOR COLOR

By default, the sunburst plot is colored by the root categorical data. If it is required that the color needs to be categorized by the hierarchical data, you need to update the marker color in the figure trace below.

```
fig.update_traces(marker_colors=color_list)
```

where the color_list can be the list of color value as you defined for each category in the plot. This is the example that the definition of color list using Python color group coding by fig object within the plot.

```
import pandas as pd

value_num_list=pd.factorize(fig.data[0].labels)[0]

color_list=[px.colors.qualitative.Set3[c] for c in value_num_list]
```

UPDATE FIGURE TRACE FOR TEXT INFORMATION

The text information inside the sunburst plot can be added in the figure trace below to display the label of each hierarchical data and the number of subjects in each hierarchical data.

```
fig = update_traces(textinfo= 'label+value',
                    texttemplate='%{label}<br> (N=%{value})',
                    insidetextorientation='auto')
```

ADD ANNOTATION FOR ALL PATHS

The annotation text can be added as well if it is needed in the plot. In Python Plotly package, annotation can be added with add_annotation coding in the following example.

```
fig.add_annotation(x=0.8, y=0.5,
                   text='Annotation Text'
                   xanchor="left",
                   font_size=10,
                   showarrow=False)
```

ADD DISCRETE LEGEND FOR EACH CATEGORY IN THE PLOT

By default, there is no categorical legend to explain the color list of the hierarchical data. If it is required to display the categorical legend with marker symbol, you can use add_shape and add_annotation for the maker symbol and text for the categorical legend.

The following coding is the example to display the categorical legend.

```
for i in range(len(color_list)):
        fig.add_shape(type="rect", x0=1.06, y0=0.975-0.0345*i, x1=1.075, y1=0.989-0.0345*i,
                      fillcolor=df_distinct_color['Category Color'].tolist()[i] )
        fig.add_annotation(x=1.1, y=1-i*0.034, text=label_list[i], font_size=9,
                      xanchor="left", align='left', showarrow=False)
```

Finally Figure 6 is the output of the sunburst plot.

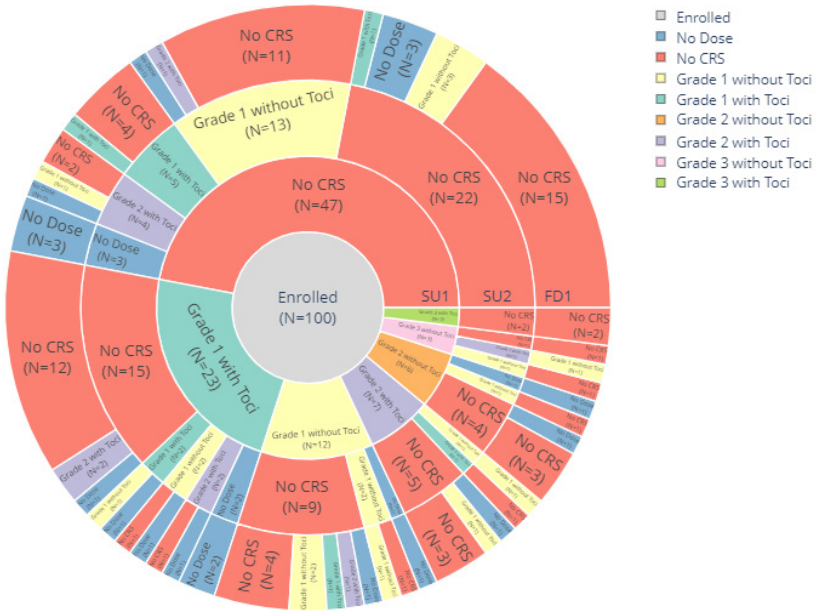

Figure 6: Sunburst Plot

SUBMISSION TO FDA WITH PYTHON

- The FDA Position Statement - Statistical Software Clarifying Statement clarifies that the “FDA does not require use of any specific software for statistical analyses, and statistical software.” All of the FDA and PMDA statements about software programs are written in a general way, they apply to any software program for analyses.

- The process and principles for submitting programs in other languages to the FDA and PMDA are the same as SAS programs. For example, Python scripts with a ".py" extension would need to be renamed to have a ".txt" extension before submitting them to the FDA, which are described in the Reviewer's Guide.
- For Python coding, the submission codes need to be defined and have comments clearly for the package information. (eg: package name, package version, etc,.) The source code or package must be accessible by all, so that FDA can review and run the Python code successfully.
- More base packages are suggested and less widely used packages are not suggested. Packages are selected based on the inner python SME group validation and programmer's experience.

CONCLUSION

There is Plotly package in both R and Python. So either R or Python can generate the sunburst plot. Currently in Year of 2023, SAS cannot generate this complex kind of sunburst plot.

If the user chooses R or Python to generate the sunburst plot analysis, there are three steps.

- Firstly, a correct dataframe structure with columns as each level data should be derived.
- Secondly, sunburst plot statement is used to generate the sunburst chart.
- Thirdly, color, text, annotation, and legend are customized.

Python is easier than R for user to customize the color group, label text, have annotation, and add categorical legend.

If the open source language needs to be submitted to the agency, the package installed needs to be clearly defined and all the source codes must be accessible.

More base packages are suggested and less widely used packages are not suggested by the instruction of your company inner open source programming SME group.

CONTACT INFORMATION

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

Name: Wei (Tony) Zhang
Enterprise: Pfizer (China) Research and Development Co.,Ltd.
E-mail: wei.zhang18@pfizer.com