PharmaSUG 2025 – OS-096

A Gentle Introduction to creating graphs in Python, R and SAS

Dane Korver, RTI International

ABSTRACT

If you have read my previous papers, I have used Wordle and temperature data to teach us how to create various graphs in SAS and R. This paper will create a bar and line chart using a very small dataset to help us grow our abilities in graphing our data using Python, R and SAS to help us visualize trends over time.

INTRODUCTION

I created a small dataset to help us practice our graphing abilities in Python, R and SAS.

Date	Attendance
June 2024	28
July 2024	21
September 2024	20
October 2024	15
November 2024	13
January 2025	68
February 2025	42
April 2025	22

Table 1. The data.

USING SAS

For SAS, there are five ODS Graphical procedures to help us create SAS graphs. The three that I use the most often are PROC SGPLOT, PROC SGPANEL and PROC SGSCATTER. For our current example, we will be using PROC SGPLOT to create our bar and line charts.

Here is a summary of the five ODS Graphical procedures for creating SAS graphs:

PROC SGPLOT	Used to create statistical graphics such as histograms, scatter plots and line plots
PROC SGPIE	Used to create statistical graphics such as pie charts and donut charts
PROC SGMAP	Identifies the data sets needed for map areas, map response values, and overlay plots
PROC SGPANEL	Used to create a panel of graphs for the values of one or more classification variables
PROC SGSCATTER	Used to create a paneled graph of scatter plots for multiple combinations of variables

To create the bar chart, we will use the **VBAR** statement and to create the line chart we will use the **SCATTER** and **SERIES** statement. To assist with determining colors for the graph, I

use colorbrewer2.org. Since my date variable is formatted as a character, I used a PROC FORMAT to ensure the categorical order of my Date in my graph.

```
proc format;
value $date_order
'06/24' = 'June2024'
'07/24' = 'July2024'
'09/24' = 'Sep2024'
'10/24' = 'Oct2024'
'11/24' = 'Nov2024'
'01/25' = 'Jan2025'
'02/25' = 'Feb2025'
'04/23' = 'Apr2025'
;
run;
```

Code to create the bar chart in SAS.

```
proc sgplot data=sample_data;
format Date $date_order.;
vbar Date/response=Attendance datalabel fillattrs=(color=CX377eb8)
outlineattrs=(color=CXa6cee3) barwidth=0.5;
xaxis label="Month" discreteorder=data labelsttrs=(Color=Black Family=Arial
Size=9);
yaxis label="Number of Attendees" grid values=(0 to 100 by 25)
labelsttrs=(Color=Black Family=Arial Size=9);
title font=arial height=12pt "Monthly Attendance";
run;
```

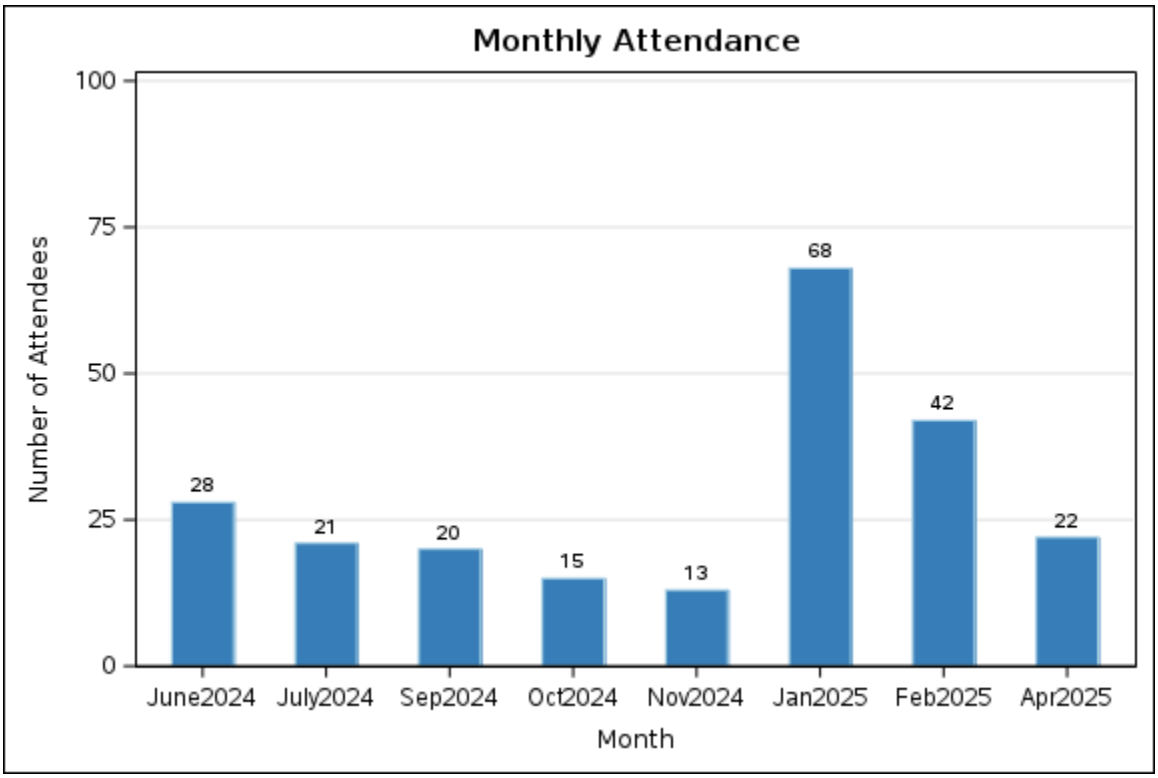

Figure 1. Bar chart using SAS

Code to create the line chart in SAS.

```
proc sgplot data=sample_data noautolegend;
format Date $date_order.;
series x=Date y=Attendance/lineattrs=(color=CXa6cee3 thickness=2);
scatter x=Date y=Attendance/markerattrs=(color=CX377eb8
symbol=circlefilled size=10);
xaxis label="Month" discreteorder=data labelsttrs=(Color=Black Family=Arial
Size=9);
yaxis label="Number of Attendees" grid values=(0 to 100 by 25)
labelsttrs=(Color=Black Family=Arial Size=9);
title font=arial height=12pt "Monthly Attendance";
run;
```

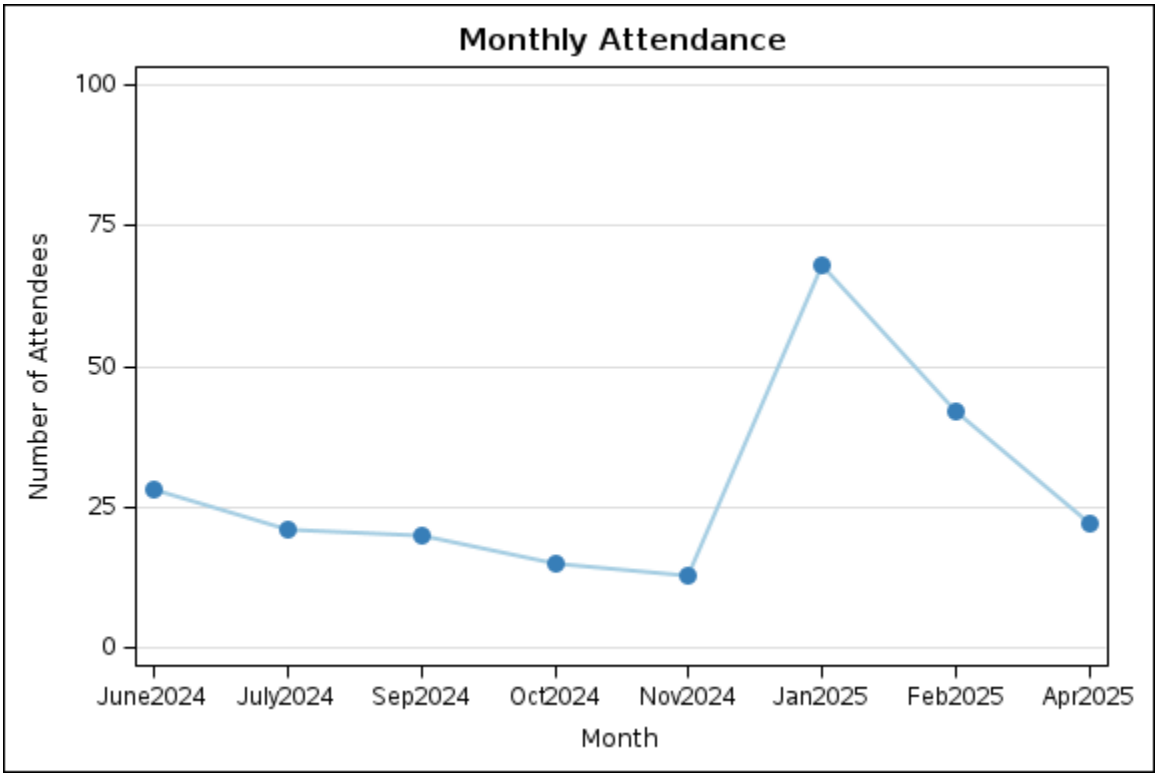

Figure 2. Line graph using SAS.

USING R

In R, the preferred method for creating a graph is to use the **ggplot2** package. Similar to creating graphs in SAS, start with the most basic graph and then add your style elements as you see fit.

Here is the basic structure of ggplot2:

```
ggplot(data = <Data>) +
<Geom_Function>(mapping = aes(<Mappings>),
stat = <Stat>,
position = <Position>) +
<Coordinate_Function> +
<Facet_Function> +
<Scale_Function> +
<Theme_Function>
```

Similar to using PROC FORMAT to ensure the categorical order of my Date variable in my graph.

```
custom_labels <- c(
  "06/24" = "June2024",
  "07/24" = "July2024",
  "09/24" = "Sep2024",
  "10/24" = "Oct2024",
  "11/24" = "Nov2024",
  "01/25" = "Jan2025",
  "02/25" = "Feb2025",
  "04/23" = "Apr2025"
)

sample_data$Date <- factor(rtsug_data$Date, levels = c("06/24", "07/24", "09/24",
"10/24", "11/24", "01/25", "02/25", "04/23"))
```

Code to create the bar chart in R.

```
ggplot(data = sample_data, aes(x = Date, y = Attendance)) +
  geom_bar(stat="identity",fill="#377eb8",color="#a6cee3",width=0.5) +
  geom_text(aes(label = Attendance), vjust = -0.5, size = 4) +
  labs(
    title = "Monthly Attendance",
    x = "Month",
    y = "Number of Attendees"
  ) +
  scale_x_discrete(labels = custom_labels) +
  scale_y_continuous(limits=c(0,100)) +
  theme_light() +
  theme(
    plot.title = element_text(hjust = 0.5),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.y = element_blank()
  )
```

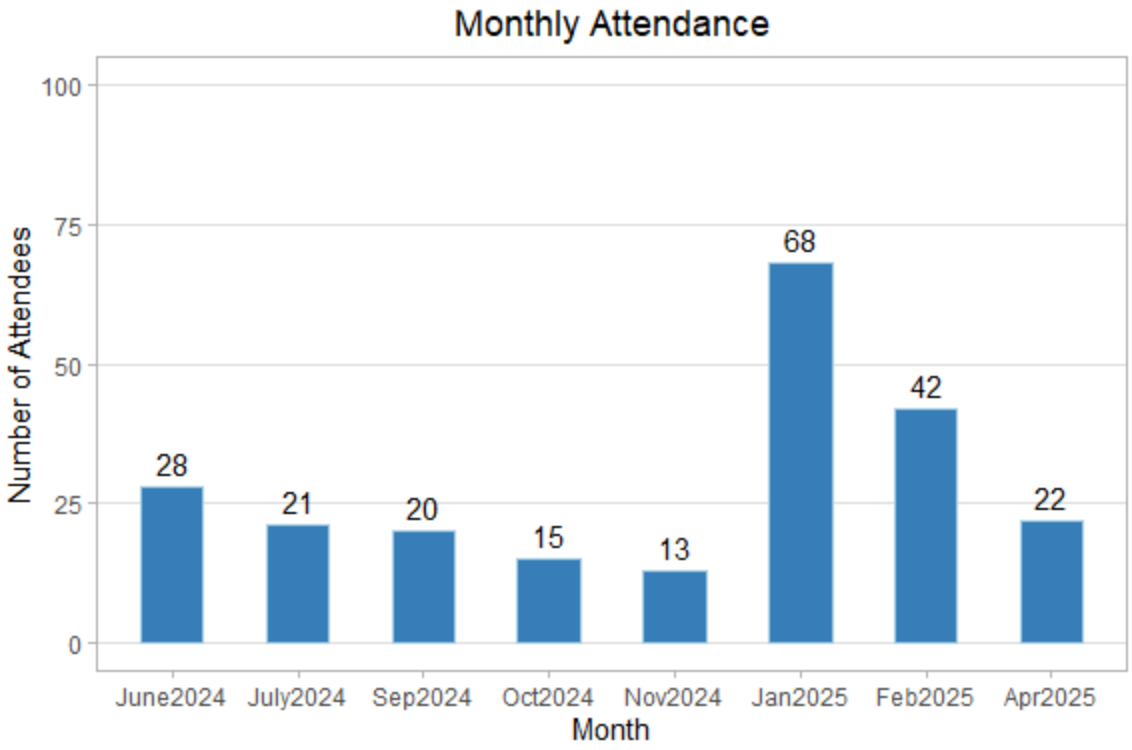

Figure 3. Bar chart using R's ggplot2

Code to create the line chart in R.

```
ggplot(data = sample_data, aes(x = Date, y = Attendance, group = 1)) +
  geom_line(color = "#a6cee3", linewidth = 3) + # Line connecting the points
  geom_point(color = "#377eb8", size = 4) + # Points on the line
  labs(
    title = "Monthly Attendance",
    x = "Month",
    y = "Number of Attendees"
  ) +
  scale_x_discrete(labels = custom_labels) +
  scale_y_continuous(limits=c(0,100)) +
  theme_light() +
  theme(
    plot.title = element_text(hjust = 0.5),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.y = element_blank()
  )
```

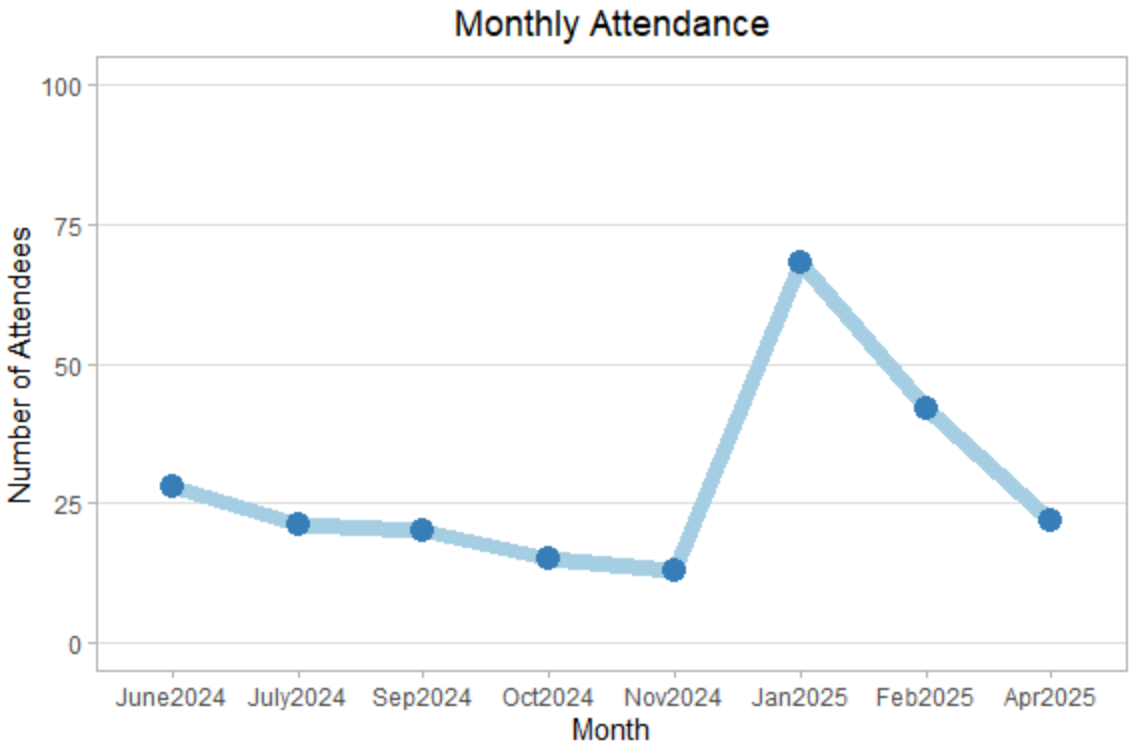

Figure 4. Line chart using R's ggplot2

USING PYTHON

In Python, there several packages available: matplotlib, seaborn, bokeh, plotly, plotnine. For this paper, we are going to be using the **matplotlib** package.

Reading in my data into Python.

```
import pandas as pd
from matplotlib import pyplot as plt

# Load the dataset into a pandas DataFrame
data = {
   "Date": ["06/24", "07/24", "09/24", "10/24", "11/24", "01/25", "02/25", "04/25"],
   "Attendance": [28, 21, 20, 15, 13, 68, 42, 22]
}
df = pd.DataFrame(data)

# Convert the Date column to datetime format
df["Date"] = pd.to_datetime(df["Date"], format="%m/%y")
print(df)
```

Code to create the bar chart in Python.

```
# Plot the bar chart
plt.figure(figsize=(8, 5))
bars = plt.bar(df["Date"].dt.strftime('%b %Y'), df["Attendance"], color='skyblue',
edgecolor='black')
plt.title("Attendance Over Time", fontsize=16)
plt.xlabel("Date", fontsize=12)
plt.ylabel("Attendance", fontsize=12)
plt.xticks(rotation=45, ha='right', fontsize=10)
plt.ylim(0, 110)
plt.yticks(range(0, 110, 10))
for bar in bars:
   height = bar.get_height()  # Get the height of each bar
   plt.text(bar.get_x() + bar.get_width() / 2, height + 1,  # Position text slightly
above the bar
            str(height), ha='center', va='bottom', fontsize=10)
plt.tight_layout()
plt.show()
```

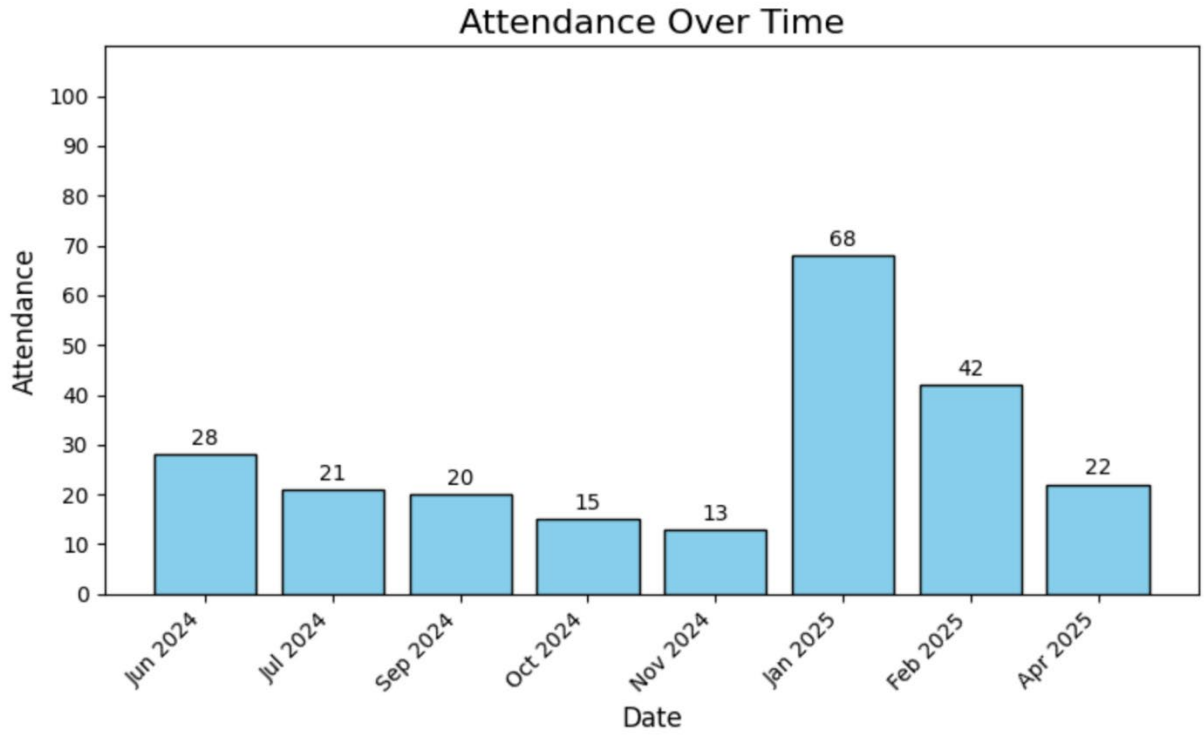

Figure 5. Bar chart using Python's matplotlib.

Code to create the line chart in Python.

```
# Plot the line graph
plt.figure(figsize=(8, 5))
plt.plot(range(len(df)), df["Attendance"], marker='o', color='skyblue', linestyle='-',
linewidth=2, label="Attendance")
plt.title("Attendance Over Time", fontsize=16)
plt.xlabel("Date", fontsize=12)
plt.ylabel("Attendance", fontsize=12)
# Set x-axis ticks to be equally spaced
plt.xticks(range(len(df)), df["Date"].dt.strftime('%b %Y'), rotation=45, ha='right',
fontsize=10)
# Set y-axis range and ticks
plt.ylim(0, 100)
plt.yticks(range(0, 110, 10))
# Add numbers on top of the data points
for i, value in enumerate(df["Attendance"]):
    plt.text(i, value + 1, str(value), ha='center', va='bottom', fontsize=10)
plt.tight_layout()
plt.show()
```

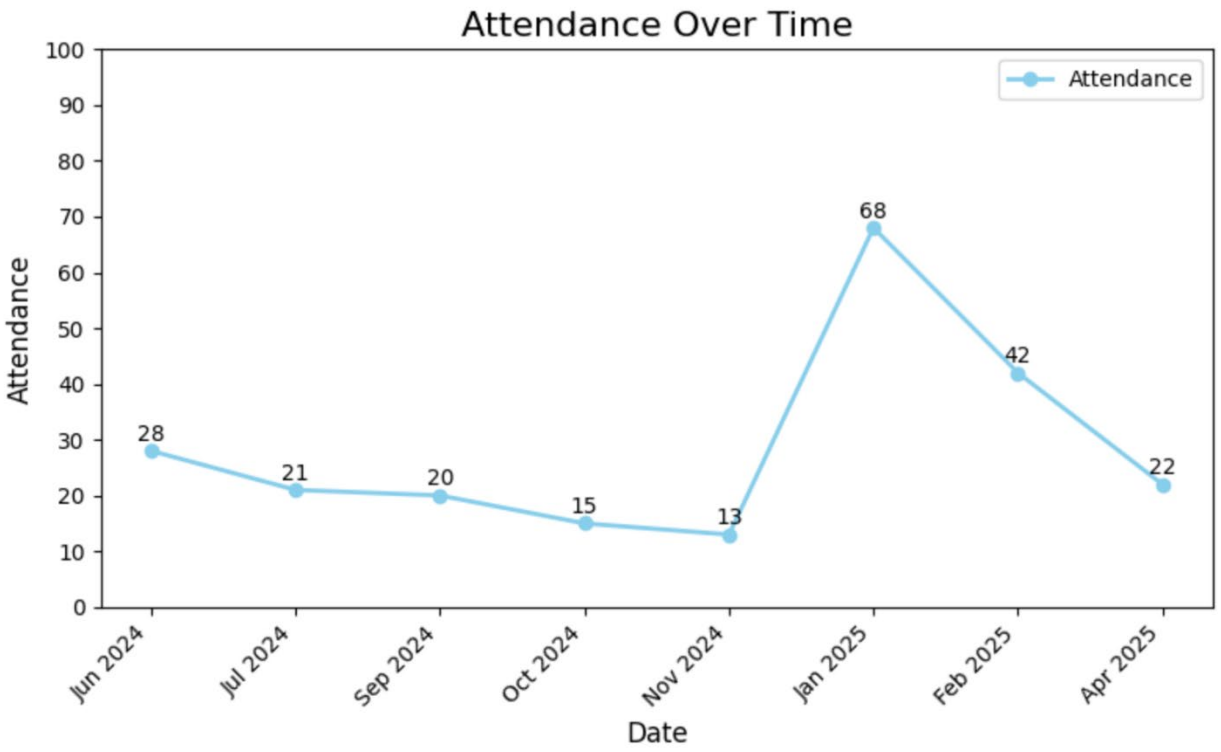

Figure 6. Line graph using Python's matplotlib.

RECOMMENDED READING

- Checkout SAS's "Graphically Speaking" blog here: https://blogs.sas.com/content/graphicallyspeaking/ or "The DO Loop" blog here: https://blogs.sas.com/content/iml/
- Harris, Kriss, and Richann Watson. 2020. SAS Graphics for Clinical Trials by Example. Cary, NC: SAS Institute Inc.
- Wickham, Hadley, et al. "ggplot2: Elegant Graphics for Data Analysis (3e)." Ggplot2, https://ggplot2-book.org/
- "Visualization with Python." Matplotlib, https://matplotlib.org/

CONTACT INFORMATION

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

Dane Korver
RTI International
dkorver AT rti DOT org

Replacing "AT" and "DOT" with the appropriate symbols.