

A Gentle Introduction to Creating SAS and R Graphs

Dane Korver, RTI Health Solutions, RTP, NC

ABSTRACT

In my previous papers, I focused on creating bar charts using SAS from my Wordle data. This paper is going to focus on learning how to create line graphs using SAS and R from weather data since I am curious about the temperature difference between Raleigh, NC and Minneapolis, MN for the year 2023. The data comes from the [National Centers for Environmental Information \(NOAA\)](#) website.

INTRODUCTION

I am curious what the graph of the average daily temperature for the year 2023 looks like for the temperatures recorded in Raleigh, NC and Minneapolis, MN. I want to find out if there was any time during the year where the graphs intersected or followed each other. This paper will demonstrate how to plot this data using SAS and R.

DATA

When I obtained the data from the National Centers for Environmental Information (NOAA) website, it was a CSV file with the following variables: Date, TAVG, TMAX, TMIN, PRCP, SNOW and SNWD. I only kept the variables: Date, TAVG, TMAX, TMIN for my purposes and filtered the data to keep the year 2023 temperatures. After I imported the data into SAS and R, I created a variable named CATEGORY to help with creating a nice label for my graph so we can identify what line belongs to Raleigh, NC or to Minneapolis, MN. I used the **MONTH** call function in SAS to extract the month part from the Date variable and then combined the Raleigh and Minneapolis weather data into one dataset. The combined dataset now has the daily weather data for Raleigh and Minneapolis. Since I am interested in the average monthly data for Raleigh and Minneapolis, I used **PROC MEANS** with an output statement so I can use the **PROC MEANS** output in my **PROC SGPLOT** to create the graph.

```
ods output summary=weather_stats;  
proc means data=WORK.WEATHER_DATA stackodsoutput mean min max;  
class Category month;  
var TAVG;  
run;
```

SAS GRAPH

In my previous papers, I used **PROC SGPLOT** with **VBAR** to create a barchart. We will continue using **PROC SGPLOT**, but with the **SERIES** statement to create a line graph. The code below creates a very simple line graph which still tells us a lot of information visually especially when we look at the two lines between May and June!

```
proc sgplot data=work.weather_stats;  
series x=month y=Mean / group=Category;  
yaxis label="Average Degrees Fahrenheit" values=(0 to 100 by 5);  
xaxis label="Month" values=(1 to 12 by 1);  
run;  
quit;
```

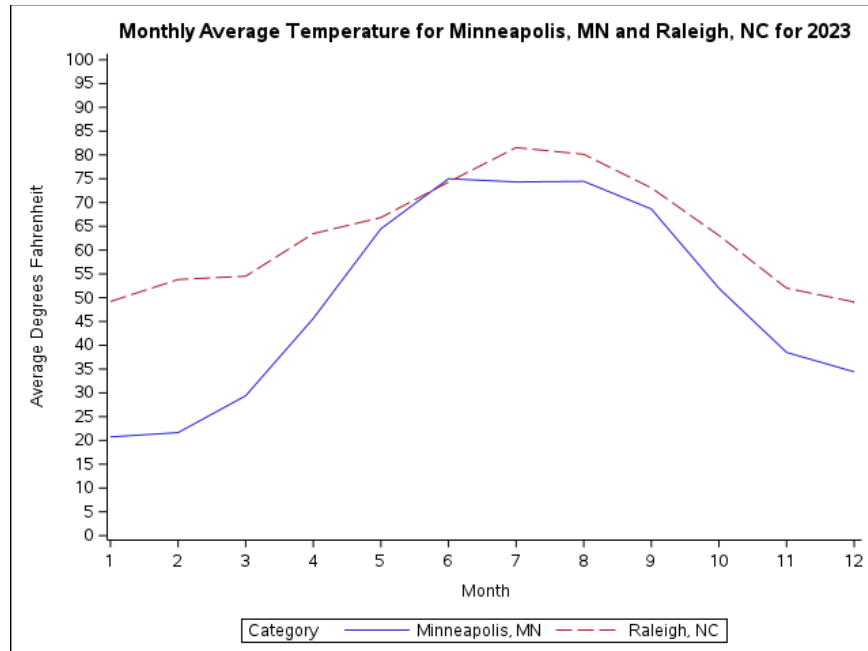


Figure 1. Basic graph using no style options

Simple is great, but we can do better! This is where we start adding in the **PROC SGPLOT** style options. First, let's ask ourselves, what enhancements could we add to this graph to make it more appealing?

Here are some ideas:

- Use the name of the month instead of the numbers
- Add a reference line for the freezing temperature (32 degrees Fahrenheit)
- Put the name of the city next to the line on the graph for easy reference
- Enhance the graph line and text
- Change the graph background

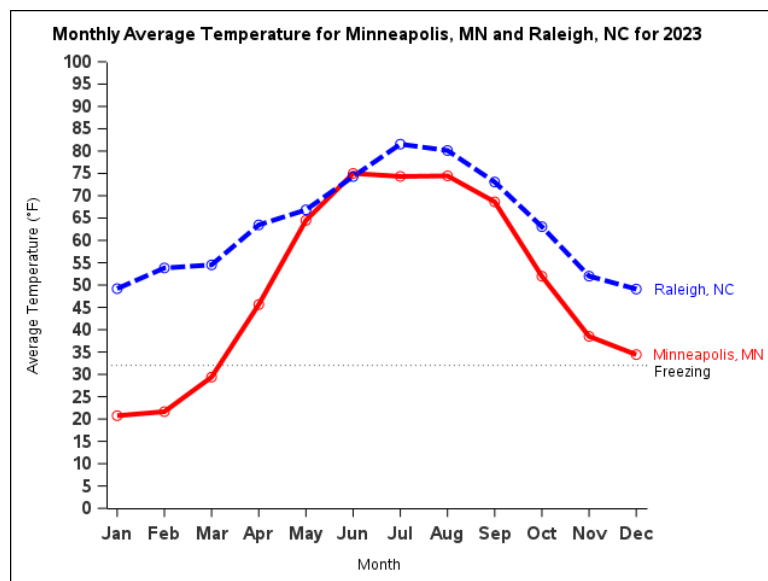


Figure 2. Fancy graph

To translate our ideas into SAS lingo, this is what I did:

- Use the name of the month instead of the numbers
 - This idea did not involve a **PROC SGPLOT** option, I simply used **PROC FORMAT** and then used the **format** statement with my **PROC SGPLOT**
- Add a reference line for the freezing temperature (32 degrees Fahrenheit)
 - Used the **REFLINE** statement
- Put the name of the city next to the line on the graph for easy reference
 - I used the **curvelabel, curvelabelpos, curvelabelloc** options with the **SERIES** statement.
- Enhance the graph lines and text
 - I used the **lineattrs, markerattrs, and valueattrs** options to satisfy my graphical desires.

R GRAPH

In R, the preferred method for creating a graph is to use the package **ggplot2**. 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>
```

Without any style elements, here is basic plot of our weather data using ggplot2:

```
ggplot(weather_data, aes(x = month, y = Mean, color = Category)) +  
  geom_line() +  
  geom_point()
```

To give our graph more "curb appeal," here is one way:

```
ggplot(weather_data, aes(x = month, y = Mean, color = Category)) +  
  geom_line(linewidth = 1.0) +  
  geom_hline(yintercept=32,linetype="dashed")+  
  geom_point()+  
  annotate(geom = "text", x = 1.5, y = 35, label = "Freezing")+  
  labs(title = "Monthly Average Temperature for",  
    subtitle = "Minneapolis, MN & Raleigh, NC for the Year 2023",  
    x = "Month",  
    y = "Average Temperature (\u00b0F)") +  
  theme_minimal() +  
  theme(plot.title=element_text(hjust = 0.5),  
    plot.subtitle=element_text(hjust = 0.5),  
    axis.line = element_line(colour = "grey50"))+  
  scale_x_continuous(breaks = 1:12) +  
  scale_y_continuous(breaks = seq(0, 100, by = 10),limits = c(0, 100))
```

Notice how each line adds a stylist element to the graph.

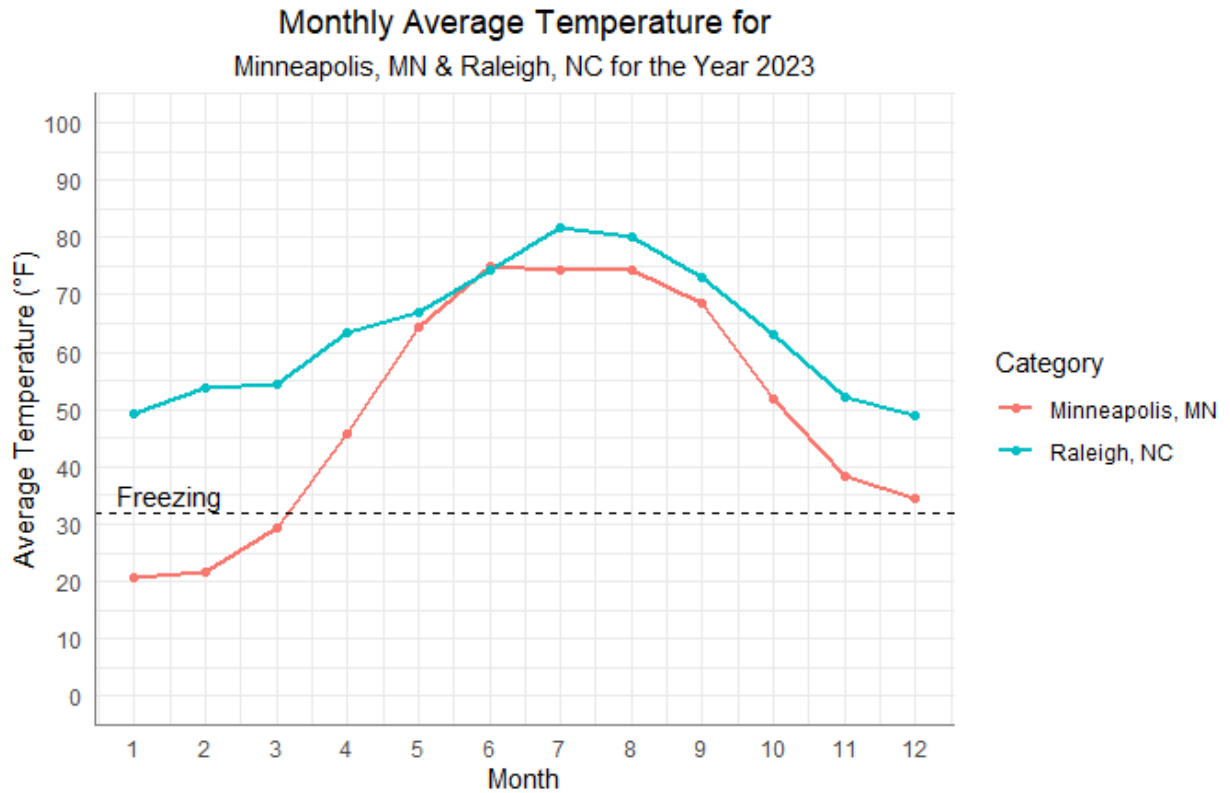


Figure 3. R graph

CONCLUSION

Besides satisfying our curiosity for what the graph of the average daily temperature for the year 2023 looked like for the temperatures recorded in Raleigh, NC and Minneapolis, MN and where the temperatures between the two locations intersected, we learned how to graph this data in SAS and R.

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/>

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.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.