Visualizing Data in Chemistry with ggplot2

Aaron Taing

October 31, 2017

Introduction

Welcome to my post on creating graphics out of data sets. I am a Chemistry major taking Stat 133 (Concept in Computing with Data) at UC Berkeley. My main interests are in analytical chemistry. Essentially, this is the field of measuring things, which of course results in data collection. I wanted to learn how to visualize data using software more powerful than Excel, and so I am taking Stat 133 and making this post about ggplot2. Since this is an assignment, code chunks will be included.

Background: ggplot2 in Stat 133

In class so far, we have learned a significant amount about the ggplot2 package. Through lecture and labs, we have learned how to make scatterplots, box plots, histograms, line plots, bar plots, and density plots. Additionally, we have learned to add enhancers such as color, faceting, and smoother lines. However, class cannot teach everything. Some topics, especially error visualization, that went nearly unmentioned are very important to scientific data.

Error in Data

When collecting data via scientific instruments, there will always be some level of uncertainty in measurements due to physical limitations. For example, in gas chromatography, the sample injector is a source of uncertainty. It cannot draw the exact same amount of volume from a sample every trial to inject into the column. The tiny fluctuations in injection volume, among other factors, cause uncertainty in each measurement, necessitating the use of multiple trials and standard deviation.

It is not just specific cases like this. Error analysis is always important when trying to draw conclusions from data. If a conclusion is not statistically significantly different from background noise/error, then you risk misleading the public with your conclusions. Due to the importance of error, I will primarily be showing how ggplot offers commands to accomplish error visualization.

```r
# loading packages needed
library(readr)
library(dplyr)
library(ggplot2)

# importing data to be used
co2.monthly <- read_csv("./data/co2-monthly-averages.csv")
congeners <- read_csv("./data/whiskey-congeners-data.csv")
```

Visualizing Measurements of Carbon Dioxide Concentration

The first data set I will look at are the monthly average measurements of atmospheric CO2 concentration taken by NOAA at the Mauna Loa Observatory (MLO) on Big Island in Hawaii [1]. The MLO is especially well known for its CO2 monitoring due to its ideal location for doing so: Hawaii is quite far from any continent, and the air sampled there is therefore less likely to be contaminated by pollution advected from nearby sources.

I said earlier that I will try to emphasize error visualization, so we should begin by trying a simple plot (Figure 1) that shows error bars. The x-axis will display time, and the labels are rotated to prevent overlapping [2]. The y-axis will display the monthly average concentration measurement. My error bars will represent standard deviation.

I looked at the error visualization commands provided by ggplot2. The command **geom_errorbar** sounds exactly like what I want, but because it places whiskers at the end of the error bars, it may be inappropriate in this plot due to the large amount of points to be plotted. The whiskers would very likely overlap and the resulting plot would probably be rather ugly. The command **geom_linerange** looks promising but since I want to plot points with an error range, it seems redundant to use both **geom_point** and **geom_linerange** when I can simply use **geom_pointrange**.

I will therefore use the command **geom_pointrange**. When plotting error in ggplot2, you need to specify the error for each point. My data set contains a column **value** for the average monthly measurements and a column **value_unc** for the uncertainty of each monthly average. Thus the range for each point can be specified by setting **ymin** to **value - value_unc** and **ymax** to **value + value_unc**. A loess line and legend [3] are also included. The code chunk is displayed below.

```r
ggplot(data = co2.monthly, aes(x = dates, y = value)) +
  geom_pointrange(aes(y = value,
                 ymax = value + value_unc, # sets the range for each point
                 ymin = value - value_unc),
                 size = 0.05, fatten = 0.05, color = "red") +

  # creating the loess line
  geom_smooth(method = "loess", size = 0.1, se = FALSE,
             aes(color = "loess line")) +

  # creating the legend while assigning colors [3]
  scale_color_manual("", values = c("loess line" = "blue")) +

  # this sets the x-axis as a date time axis, with labels 30 months apart
  scale_x_date(date_breaks = "30 months", date_labels = "%m-%Y") +

  # this rotates the x-axis date labels [2]
  theme(axis.text.x = element_text(angle = 90)) +

  labs(x = "Time (Month-Year)", y = "Concentration (parts per million)",
      title = "Monthly Average CO2 Concentration in the Atmosphere Over Time",
      caption = "Figure 1: Plotting standard deviation with geom_pointrange")
```

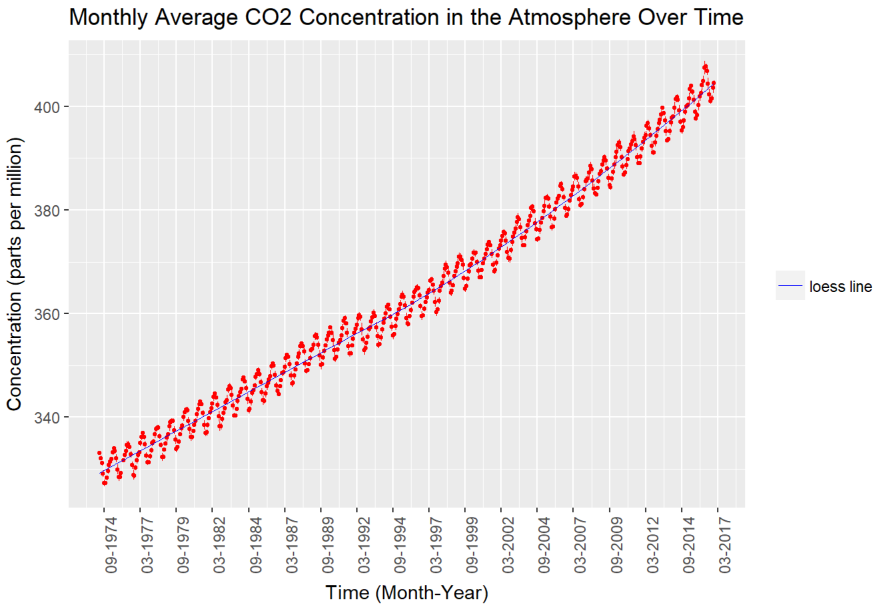

Figure 1: Plotting standard deviation with geom_pointrange

This seems pretty decent. However, I found an even nicer (in my opinion) way to display the error range by using the command **geom_ribbon**. With this command, I can specify a value range in the same exact way. However, now I can also fill in the area bound by the specified range. In the plot (Figure 2), I will fill it with the same color as the points but with more transparency.

```
# same code as before but using geom_ribbon instead
ggplot(data = co2.monthly, aes(x = dates, y = value)) +
  geom_point(size = 0.1, aes(color = "monthly averages")) +
  geom_smooth(method = "loess", size = 0.1, se = FALSE,
              aes(color = "loess line")) +
  scale_color_manual("", values = c("monthly averages" = "red",
                                    "loess line" = "blue")) +

# now using geom_ribbon
  geom_ribbon(aes(ymin = value - value_unc, ymax = value + value_unc,
                  fill = "uncertainty"), # naming the filled region for the legend
              color = NA, # this removes borders placed on the filled region
              alpha = 0.1) + # makes filling transparent
  scale_fill_manual("", values = "red") + # making the fill color red

  scale_x_date(date_breaks = "30 months", date_labels = "%m-%Y") +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(x = "Time (Month-Year)", y = "Concentration (parts per million)",
       title = "Monthly Average CO2 Concentration in the Atmosphere Over Time",
       caption = "Figure 2: Plotting standard deviation with geom_ribbon")
```

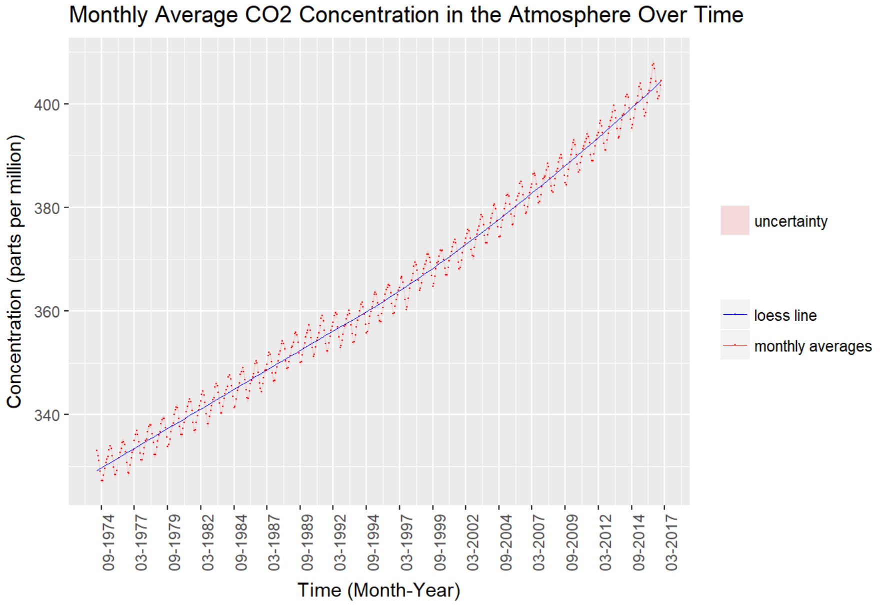

Figure 2: Plotting standard deviation with geom_ribbon

Admittedly, it is somewhat difficult to distinguish between the two graphs just now generated because the filled region is barely visible. However, this will now allow me to explain two things with one additional graph. I will create a truncated version of NOAA's data set that will span a time length from January 1976 to December 1976. The resulting plot (Figure 3) will thus have only 12 points, and it will be much easier to see the result of my code. Additionally, a one-year time span will allow me to explain the periodic rise and fall of CO2 concentrations.

```
# subset of data that goes from Jan 1976 to Dec 1976
trunc.co2 <- co2.monthly[20:31, ]

# the same exact code but applied to the truncated data set
# loess line is also removed this time
ggplot(data = trunc.co2, aes(x = dates, y = value)) +
  geom_line(size = 0.1, aes(color = "monthly average")) +
  scale_color_manual("", values = c("monthly average" = "red")) +
  geom_ribbon(aes(ymin = value - value_unc, ymax = value + value_unc,
                  fill = "uncertainty"), color = NA, alpha = 0.1) +
  scale_fill_manual("", values = "red") +
  scale_x_date(date_breaks = "1 month", date_labels = "%m-%Y") +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(x = "Time (Month-Year)", y = "Concentration (parts per million)",
       title = "Monthly Average CO2 Concentration in the Atmosphere Over One Year",
       caption = "Figure 3: Plotting standard deviation with truncated CO2 data set")
```

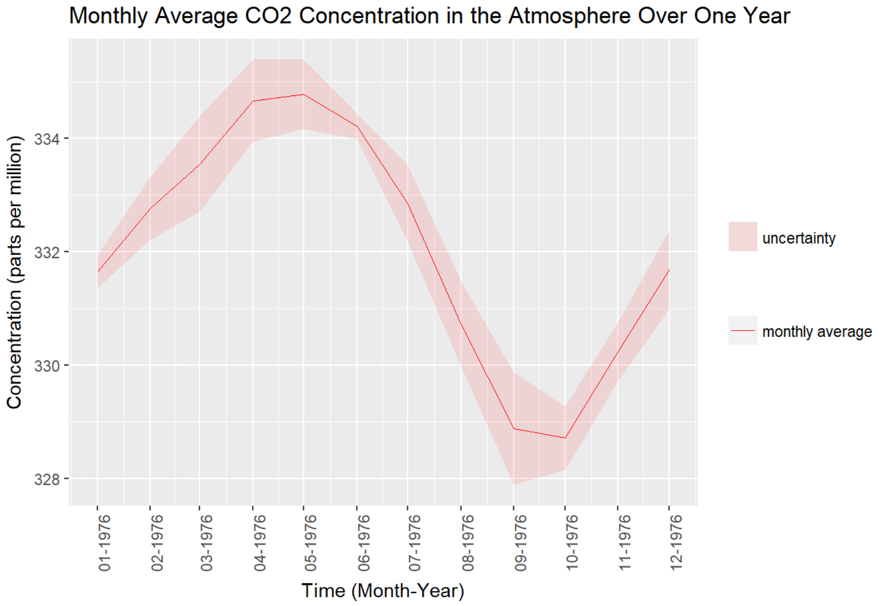

Figure 3: Plotting standard deviation with truncated CO2 data set

The effects of **geom_ribbon** are now far more visible. Now, as we follow the line generated through connecting the points, we also know the uncertainty of the line's position.

This new plot also displays the seasonal variance associated with CO2 concentration. Though we can see a clear gradual rise in atmospheric CO2 concentrations through the loess line in Figure 1 and Figure 2, there is also a periodicity associated with seasons. Beginning in Spring, plants regain their leaves, and the amount of photosynthesis rises greatly, decreasing global CO2 levels. Beginning in Fall, plants begin to lose their leaves, and photosynthesis drops while decomposition rises, increasing global CO2 levels. Of course, the seasons are opposite in the Northern and Southern Hemispheres, but air mixing between hemispheres takes place on a very large time scale. Thus, the Mauna Loa Observatory will observe the beginning of CO2 decreases around the beginning of Northern Hemisphere Spring.

Now that we have seen how nicely **geom_ribbon** can visualize error, I want to spend a little more time looking at the command I dismissed so quickly at the beginning of the post: **geom_errorbar**. Though it would make a mess in a line plot with many points, it seems like it would be a very good tool for visualizing error in a bar plot. That brings us to my next topic.

Viualizing Measurements of Congener Concentrations in Whiskey

The second data set we will look is a series of measurements I collected in a different class. In Chem 105 (Instrumental Methods in Analytical Chemistry), we used a gas chromatograph equipped with a flame ionization detector to measure the concentrations of various congeners, or fermentation byproduct, in various whiskey brands. The experiment followed a previously planned procedure [5] with similar data results. The data was processed prior to import. It contains the concentration measurement for each congener and an uncertainty value (standard deviation across three trials each) for each measurement.

I wanted to make my bar plot grouped by congener and with different color bars representing the three different whiskey brands I analyzed. The y-axis would then be the concentration of each congener in each whiskey. In this case, I need to specify the position that the bars take because the default value is for bars to stack on top of each other if grouped together [6]. Finally, I can use **geom_errorbar** so that we can finally see what the error bars look like. However, the error bars have to be adjusted in position as well because they simply appear on top of each other otherwise [7]. The result is Figure 4.

```
# here I specify dodge as how much I want the bars to separate by
# since they're grouped together, their width is 0.9, so I dodge by 0.9
dodge <- position_dodge(width = 0.9)

# this is the error I want to plot
limits <- aes(ymax = congeners$ppm + congeners$unc,
              ymin = congeners$ppm - congeners$unc)

ggplot(data = congeners, aes(x = compound, y = ppm, fill = whiskey)) +
  geom_bar(stat = "identity", position = dodge) +
  geom_errorbar(limits, width = 0.3, position = dodge) +
  theme_bw() +
  labs(x = "Congener", y = "Concentration (parts per million)",
       title = "Concentrations of Four Congeners in Three Whiskeys",
       caption = "Figure 4: Plotting a barplot with standard deviation dodge")
```

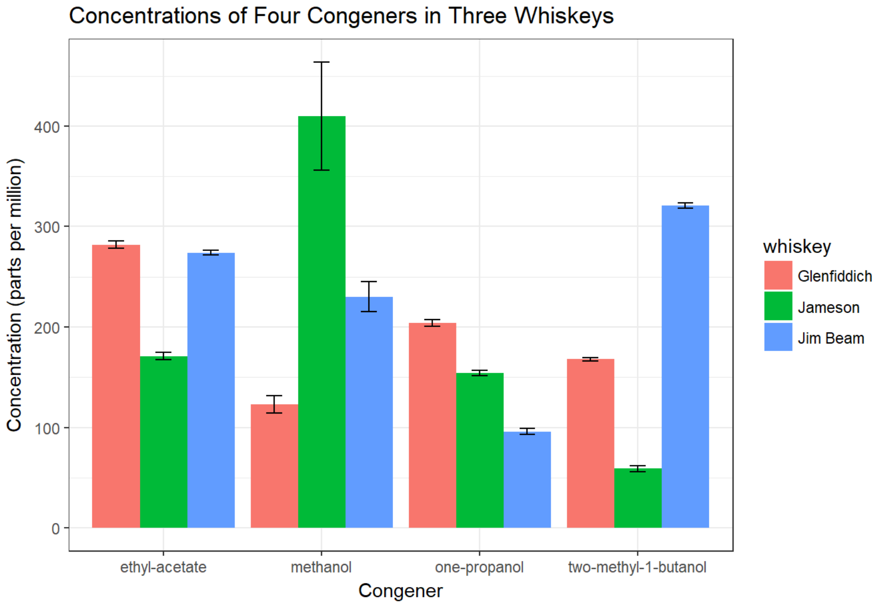

Figure 4: Plotting a barplot with standard deviation dodge

So now I have hopefully done error bars some justice.

Finishing Thoughts

I hope that my post has successfully conveyed the importance of visualizing error. When the public sees figures published, they need to be able to see error: either the standard deviations or the confidence intervals (specified at a certain percent). Otherwise, they could be mislead by the figures into drawing incorrect conclusions.

I hope you enjoyed reading the post, and to encourage proper use of data visualization and error visualization, I will end on a phrase popularized by Mark Twain.

> There are three kinds of lies: lies, damned lies, and statistics.

References

1. NOAA ESRL Global Monitoring Division. 2016, updated annually. Atmospheric Carbon Dioxide Dry Air Mole Fractions from quasi-continuous measurements at Mauna Loa, Hawaii. Compiled by K.W. Thoning, D.R. Kitzis, and A. Crotwell. National Oceanic and Atmospheric Administration (NOAA), Earth System Research Laboratory (ESRL), Global Monitoring Division (GMD): Boulder, Colorado, USA. Version 2017-8 at http://dx.doi.org/10.7289/V54X55RG.
2. https://stackoverflow.com/questions/1330989/rotating-and-spacing-axis-labels-in-ggplot2
3. https://stackoverflow.com/questions/10349206/add-legend-to-ggplot2-line-plot
4. https://stackoverflow.com/questions/29743503/how-to-add-shaded-confidence-intervals-to-line-plot-with-specified-values
5. Rice, Gary W., *J. Chem. Ed*. **1987**, *64*, 1055-1056. Path: http://digicoll.library.wisc.edu/cgi-bin/JCE/JCE-idx?type=article&did=JCE.JCE06412.i0048&id=JCE.JCE06412&isize=M
6. https://stackoverflow.com/questions/34889766/what-is-the-width-argument-in-position-dodge
7. https://www.r-bloggers.com/building-barplots-with-error-bars/