Exploring Lowess Line in Depth

Sierra Park

10/20/2017

```
#load library
library(ggplot2)
```

Introduction

“Include a lowess smooth line in the plot.”

In the beginning of the course, one of the tasks in Homework 1 was to create a LOWESS line along with a regression line through a set of points. I didn’t know what lowess smoothing was for, so I just googled the code for lowess and used it directly without understanding how it worked. For this post, I would like to go back to that portion of the assignment and relearn what lowess smooth curve is and understand how to use R function to graph it. Let us explore this area more in depth.

What is Lowess?

To start off with the definition, LOWESS stands for Locally Weighted Scatterplot Smoothing, and it is sometimes called LOESS, which stands for Locally Weighted Smoothing.

A simple “lowess/loess” curve is constructed using the “lowess()” function that finds a fitted value for each data point, connecting with lines. To find the fitted value for one data point, call it x, we find the data nearest to x and find the weighted mean. If the desired distance around x is small, we will not be able to catch sufficient data points to achieve high accuracy. Hence, we are left with a large variance. On the other hand, if the desired diefference around x is too large, then the regression may be oversmoothed, resulting in large bias.

In fact, the tradeoff between variance and bias is interesting to observe. Depending on the degree of the polynomial selected to smooth the data, the two have an inverse relationship. For example, a higher degree polynomial can predict more accurately, which lowers bias, yet since the scope of observed points were small, there may be greater variance. The typical function used is the quadratic functions since any functions of higher degree than that do not provide better results.

Use of Lowess Smoothing

So when and why do we use lowess you may ask? Lowess is a widely used tool in regression analysis to create a smooth line through a timeplot or a scatterplot to see the relationship between the variables and define a pattern or a trend. This method is particularly helpful when there are noisy data values or sparse data points that weaken the relationships between variables and interfere the line of the best fit. Hence, lowess smoothing is useful in linear regression where the least squares line is not a good representative of the data points.

You can say that the advantage of using a lowess function is that there is no need to specify a global function of any form to fit a model to the data. All we are trying to do is to fit segments of the data. However, one disadvantage is that computing lowess line is very complicated.

A real life example of the lowess line is to examine elections and voting behavior.

Difference Between Lowess and Regression

Then what is the difference between a lowess and regression line? In simple terms, a linear model fits a straight line through a set of points (the best line of fit) whileas a lowess line fits a more complicated model through a set of points baed on weighted regression.

Example of Lowess

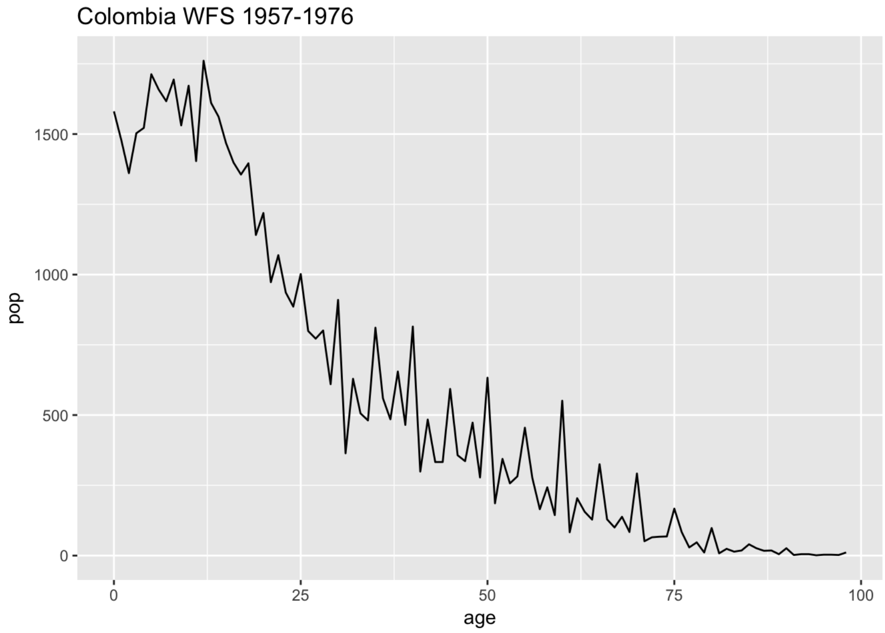
In this part, I am using an example from a website from Princeton University. Our goal is to observe the data from the Colombia WFS Household Survey 1975-1976.

```
#Read data and name it "colombia"
colombia <- read.table("http://data.princeton.edu/eco572/datasets/cohhpop.dat", col.names=c("age","pop"), header=F
ALSE)
head(colombia)
```

```
##   age  pop
## 1   0 1581
## 2   1 1477
## 3   2 1361
## 4   3 1503
## 5   4 1522
## 6   5 1713
```

To see the the general trend in age vs. population, we will create ggplot.

```
#plotting the table to get an idea of what the data look like
ggplot(colombia, aes(age, pop)) + geom_line() + ggtitle("Colombia WFS 1957-1976")
```



From the ggplot, we can see that there is a general trend: as the age increases, the population of that age decreases. However, note that the graph is not in the form of a smooth line despite of the obvious trend that we can identify.

In order to smooth this curve, the best method is to use weighted means, or we could even use the k-nearest neighbors for each point to fit a regression line to the points in the neighborhood and predict a smoother value for the index of observation. The reason why we look at the points closest to x is that the points near x are more likely to be related to each other in a simple way than points that are far apart. However, without having to mannually implement this method, we can use lowess smoother that will directly plot the smooth line.

The R base lowess function is "lowess()," and the parameter "degree" controls the degree of the polynomial, with default as 2 for quadratic. Number of iterations is controlled by the parameter "iter."

The user an specify the "bandwidth" or "smoothing parameter" that determines how much of the data is used to fit each local polynomial. The smoothing parameter, call it q, is:

$$\frac{(d+1)}{n} < q < 1$$

where d = degree of the local polynomial.

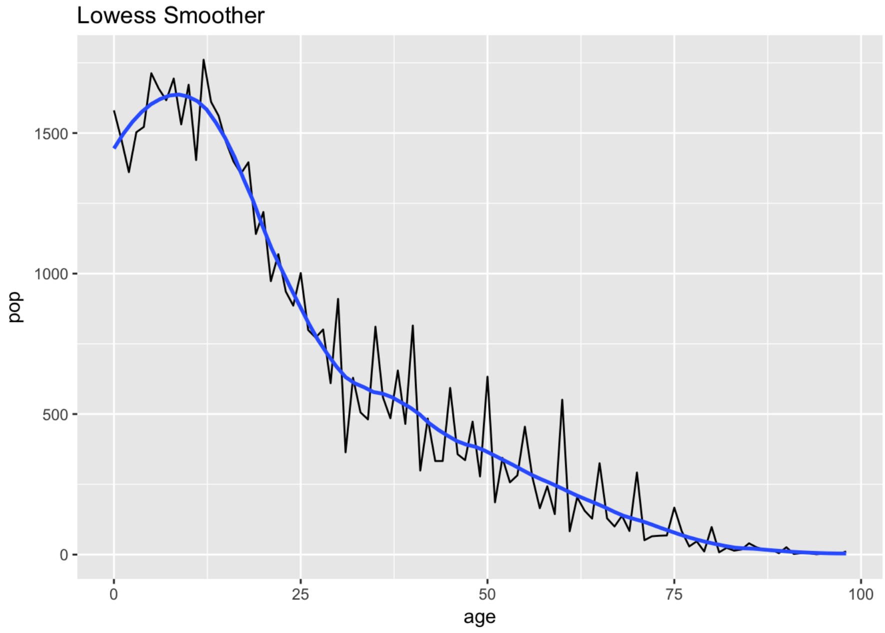
In fact, q controls the flexibility of the lowess regression function. Large q-value means that the function is smooth and is less responsive to fluctuations in the data, while small q-value means that the regression function will conform to the data.

For more information on how to access the "lowess()" function, use "?lowess." In ggplot2, however, the equivalent of lowess() is "geom_smooth()."

Now, graphing the lowess line using ggplot, we see a smoother curve that follows the general trend of the line graph.

```
#Smooth the geom_line(), which is a line graph
ggplot(colombia, aes(age, pop)) + geom_line() + geom_smooth(span=0.25, se=FALSE) + ggtitle("Lowess Smoother")
```

```
## `geom_smooth()` using method = 'loess'
```

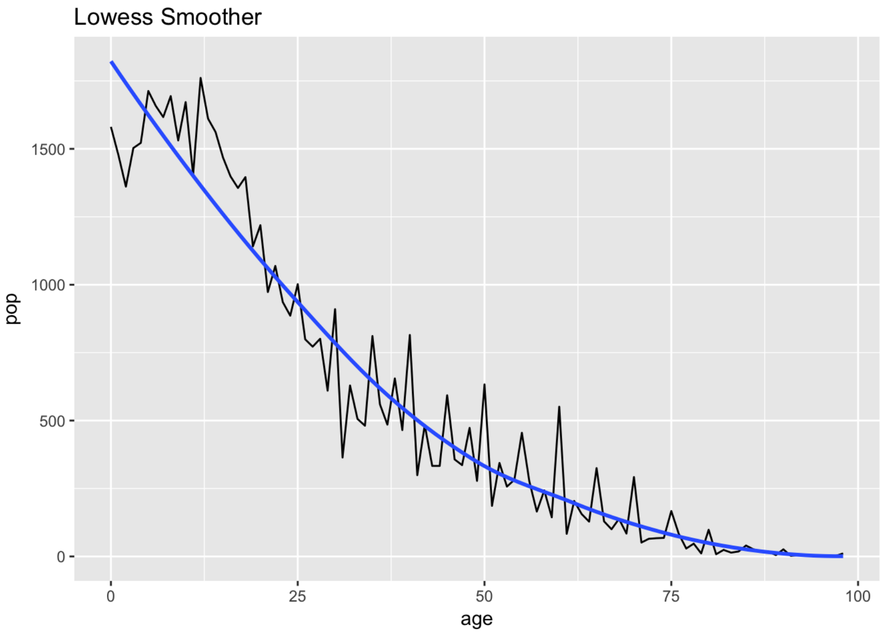


Notice that the argument “span” inside geom_smooth() is the smoothing parameter. Right now, it is at 0.25, but what happens if we increase it to 0.9?

This is the new graph below:

```
ggplot(colombia, aes(age, pop)) + geom_line() + geom_smooth(span=0.9, se=FALSE) + ggtitle("Lowess Smoother")
```

```
## `geom_smooth()` using method = 'loess'
```

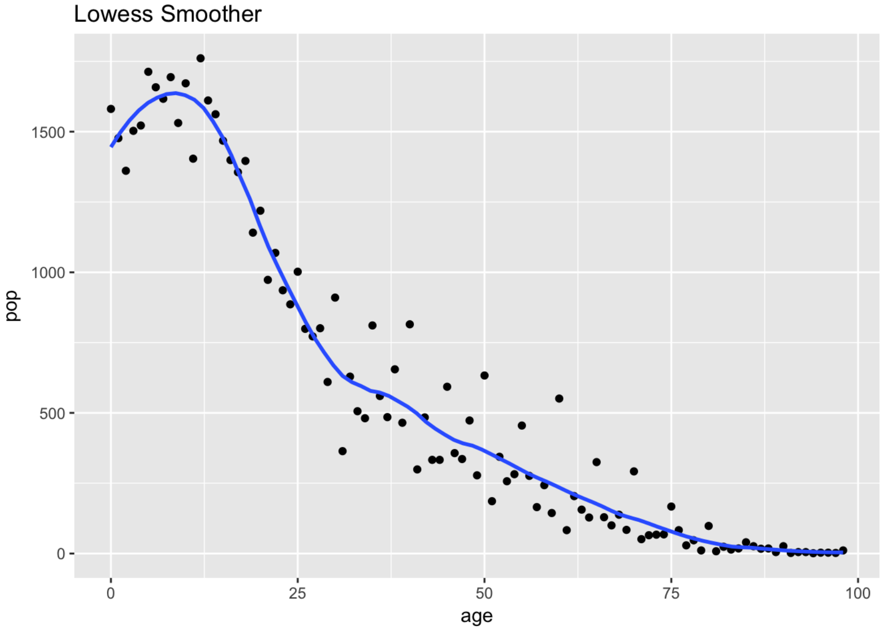


The line looks too smooth, like a straight line if the smoothing parameter had reached 1.

We looked at the lowess line vs a line plot, but let’s observe how the graph looks like if we graphed lowess line vs scatterplot:

```
#Smooth the geom_point(), which is a scatterplot
ggplot(colombia, aes(age, pop)) + geom_point() + geom_smooth(span=0.25, se=FALSE) + ggtitle("Lowess Smoother")
```

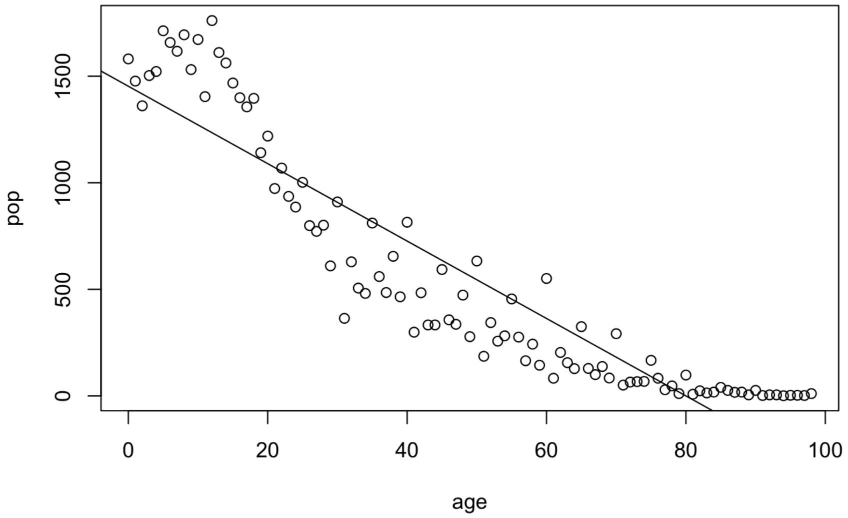
```
## `geom_smooth()` using method = 'loess'
```



It seems like the lowess line follows the scatterplot better than the line plot just visually.

Nevertheless, with using regression, we get a straight line with a negative slope that does not seem to fit the data too well:

```
#scatterplot of the dataset with linear regression line
plot(colombia$age, colombia$pop, xlab = "age", ylab = "pop")
abline(lm(colombia$pop~colombia$age))
```



Since a straight line is not flexible in accomodating the varying points of the scatterplot, this line of the best fit is not suitable, and we conclude that the lowess line is a better representation of the data.

Conclusion

In the end, the message of this post is to understand that sometimes, a better method of seeing the relationship between variables is to use a lowess line that uses a weighted average instead of using a simple linear model.

Sources:

1. http://www.statisticshowto.com/lowess-smoothing/
2. http://data.princeton.edu/eco572/smoothing1.html
3. http://geog.uoregon.edu/bartlein/old_courses/geog414f03/lectures/lec05.htm
4. https://www.statsdirect.com/help/nonparametric_methods/loess.htm
5. https://www.statmethods.net/advgraphs/axes.html
6. http://polisci.msu.edu/jacoby/icpsr/regress3/lectures/week4/15.Loess.pdf
7. http://www.itl.nist.gov/div898/handbook/pmd/section1/pmd144.htm