BlogEntry1

Zachary Palmore

5/17/2021

The Normality Assumption

In linear regression we often attempt to predict based on some data that represents reality. For this week's post, we will take a look at how to assess an important assumption that must be made to predict accurately using linear regression. To understand how to assess normality we use data from an irrigation experiment in which a sprayer nozzle delivers a certain amount of water per trial. Our goal is to estimate the amount of water delivered per trial and predict how much will be delivered at a 20, 50, and 100 trials of this sprayer nozzle. A sample of this data is provided below.

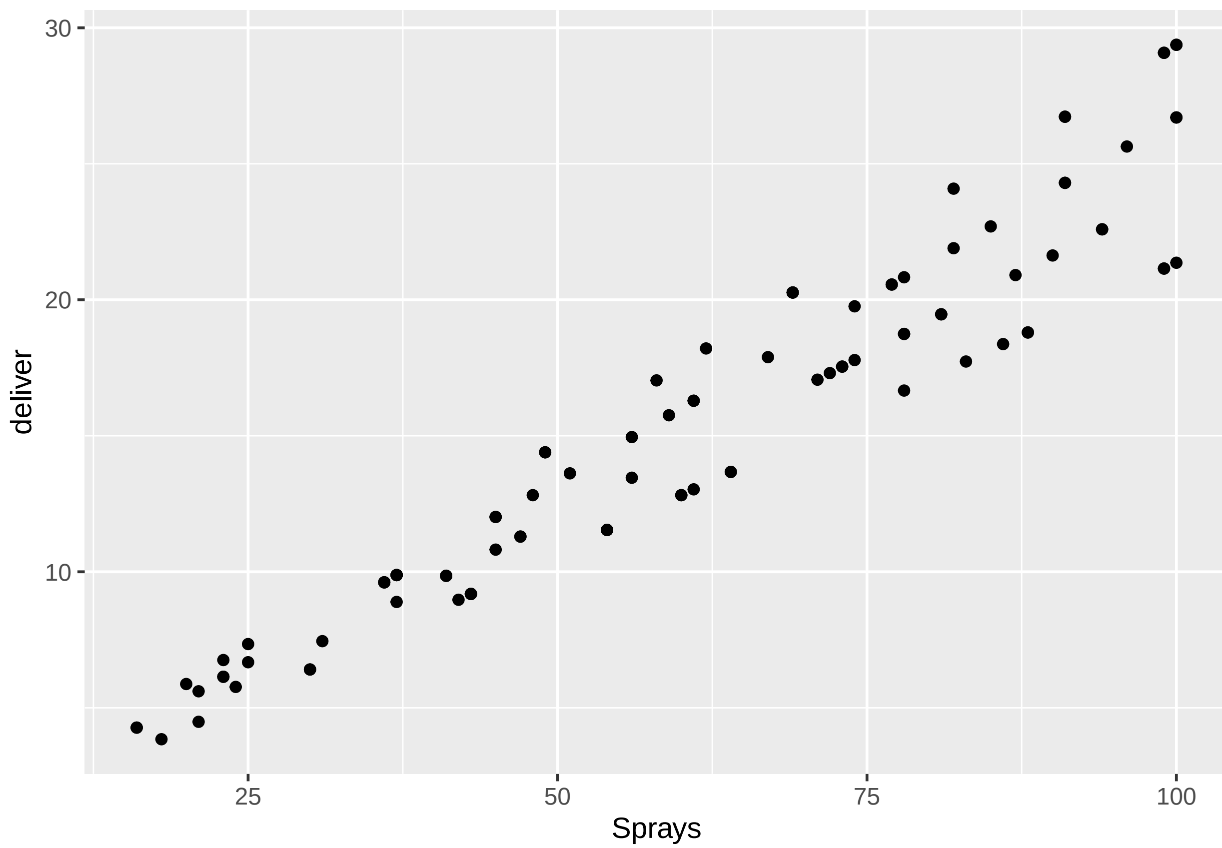
```r
library(tidyverse)
sprays <- read.csv("https://raw.githubusercontent.com/palmorezm/misc/master/Sprays.csv")
vol <- sample_n(sprays, 100, replace = T)
colnames(vol) <- c("Trial","Sprays","Diameter","Height")
vol <- vol %>%
  mutate(volume = pi*(Diameter)*Height) %>%
  mutate(deliver = volume/1000)
```

Now, this experiment was set up such that each trial contains a certain number of sprays. In this case, each trial is an independent event as the delivery of water from one nozzle does not affect the delivery of water from another. Based on this sample, we do not know which nozzle is spraying but remember our goal is to find an average for all the nozzles and predict how much water will be delivered in various intervals over the area the nozzles cover. This area is held constant for each nozzle at 8.5 centimeters in diameter. To get the volume of water, we use a formula for the volume of a cylinder:

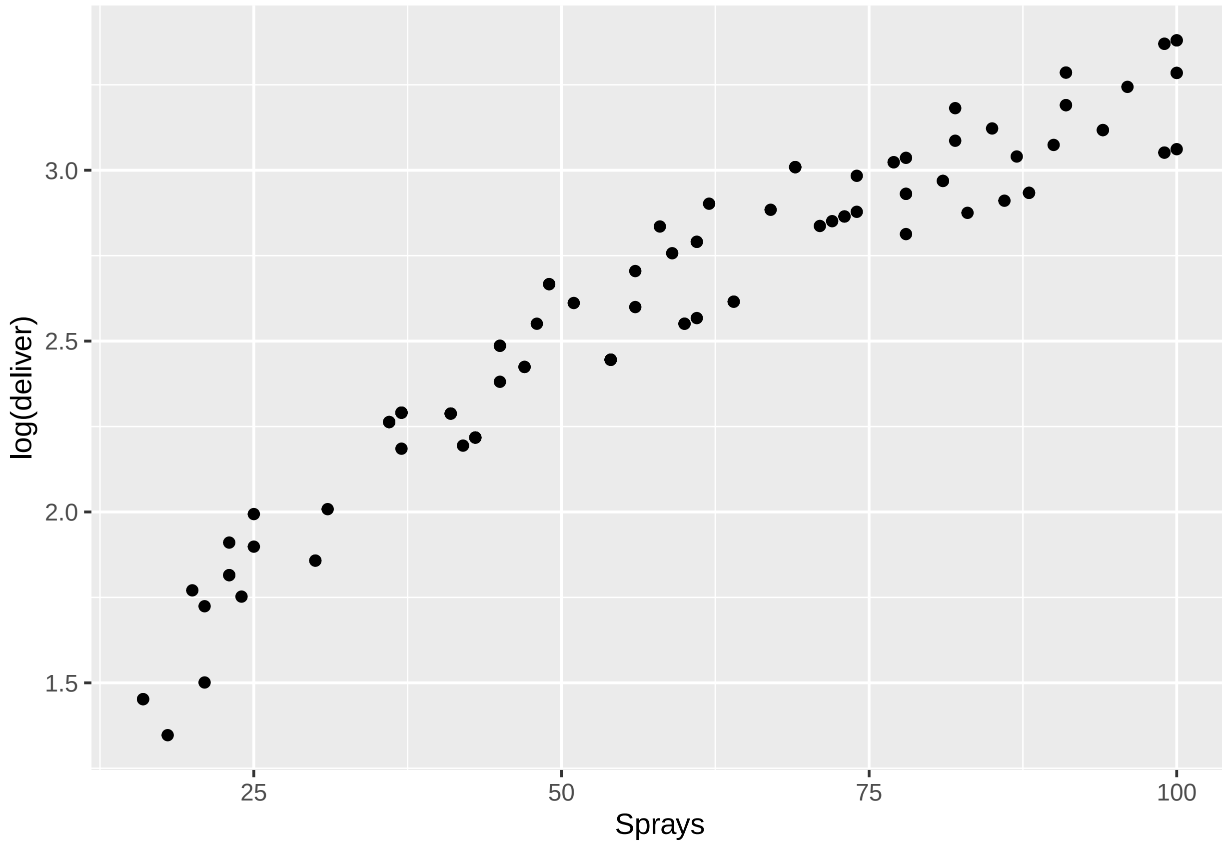
$$Volume = \pi * r^2 * h$$

Normality, along with three other assumptions of independent observations, linearity, and homoscedasticity, allows us to take advatange some statistical principles to better intepret future data points.

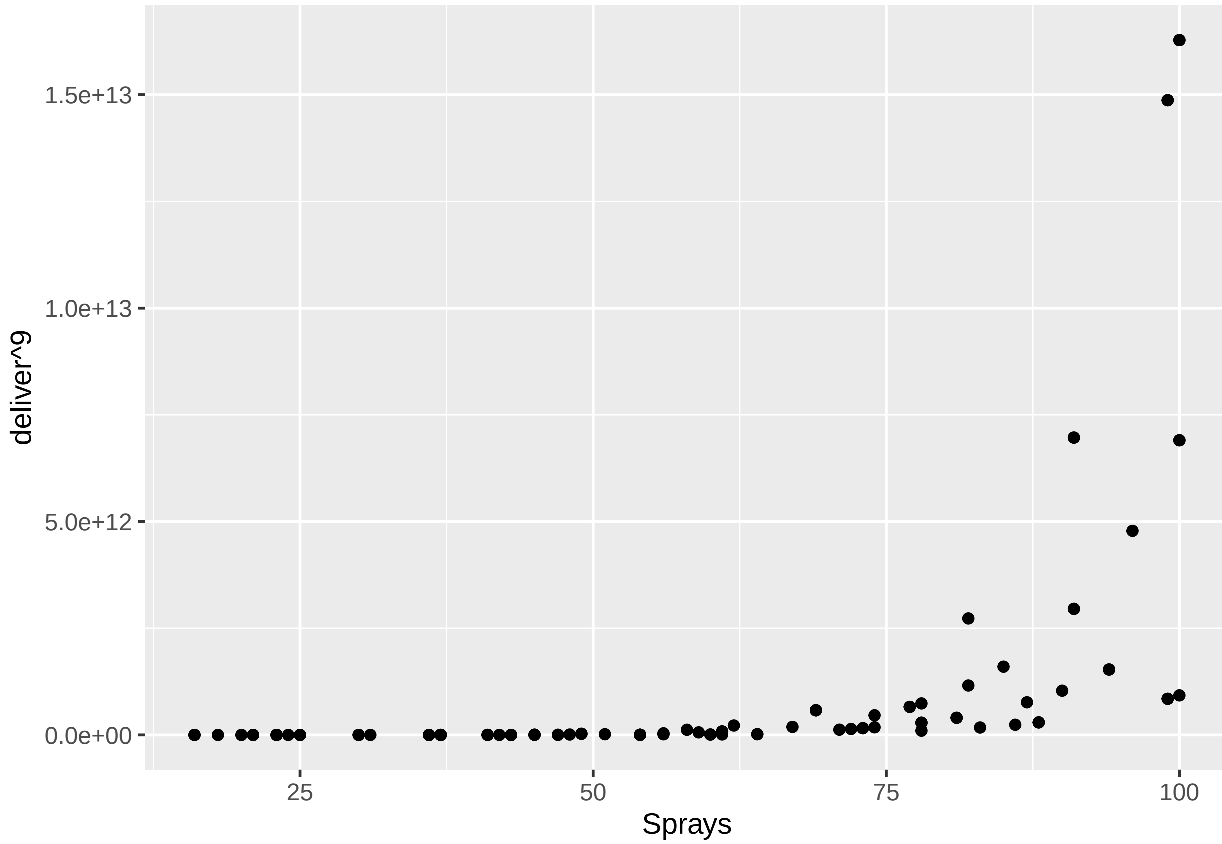
```r
# Expectation
vol %>%
  ggplot(aes(Sprays, deliver)) + geom_point()
```




```r
# Nonlinear, not normal?
vol %>%
  ggplot(aes(Sprays, log(deliver))) + geom_point()
```




```r
vol %>%
  ggplot(aes(Sprays, deliver^9)) + geom_point()
```


1.5e+13
1.0e+13
5.0e+12
0.0e+00
deliver^9
25
50
75
100
Sprays