STATS

CALCULATING Z-SCORES [WITH R CODE]

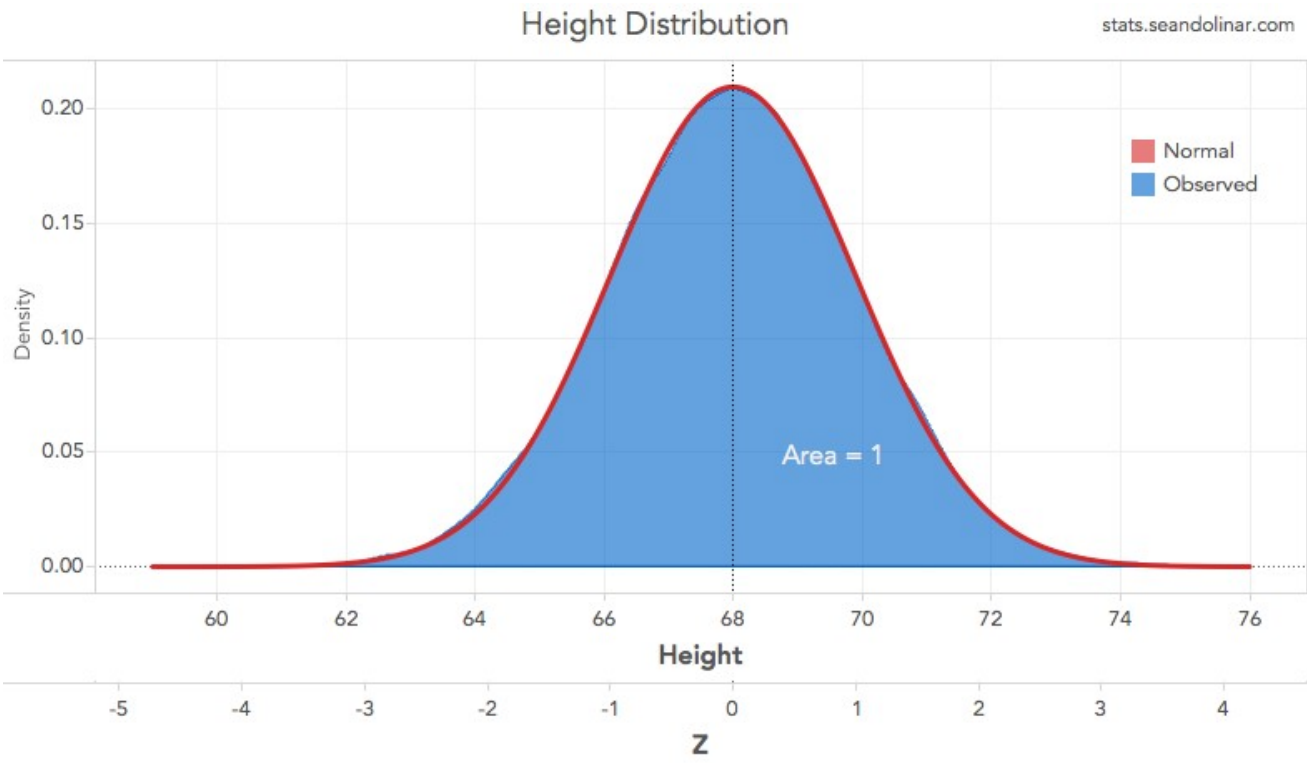
DECEMBER 17, 2014 | SEAN DOLINAR

I've included the full R code and the data set can be found on UCLA's Stats Wiki

Normal distributions are convenient because they can be scaled to any mean or standard deviation meaning you can use the exact same distribution for weight, height, blood pressure, white-noise errors, etc. Obviously, the means and standard deviations of these measurements should all be completely different. In order to get the distributions standardized, the measurements can be changed into z-scores.

Z-scores are a stand-in for the actual measurement, and they represent the distance of a value from the mean measured in standard deviations. So a z-score of 2.0 means the measurement is 2 standard deviations away from the mean.

To demonstrate how this is calculated and used, I found a height and weight data set on UCLA's site. They have height measurements from children from Hong Kong. Unfortunately, the site doesn't give much detail about the data, but it is an excellent example of normal distribution as you can see in the graph below. The red line represents the theoretical normal distribution, while the blue area chart reflects a kernel density estimation of the data set obtained from UCLA. The data set doesn't deviate much from the theoretical distribution.



The z-scores are also listed on this normal distribution to show how the actual measurements of height correspond to the z-scores, since the z-scores are simple arithmetic transformations of the actual measurements. The first step to find the z-score is to find the population mean and standard deviation. It should be noted that the sd function in R uses the sample standard deviation and not the population standard deviation, though with 25,000 samples the different is rather small.

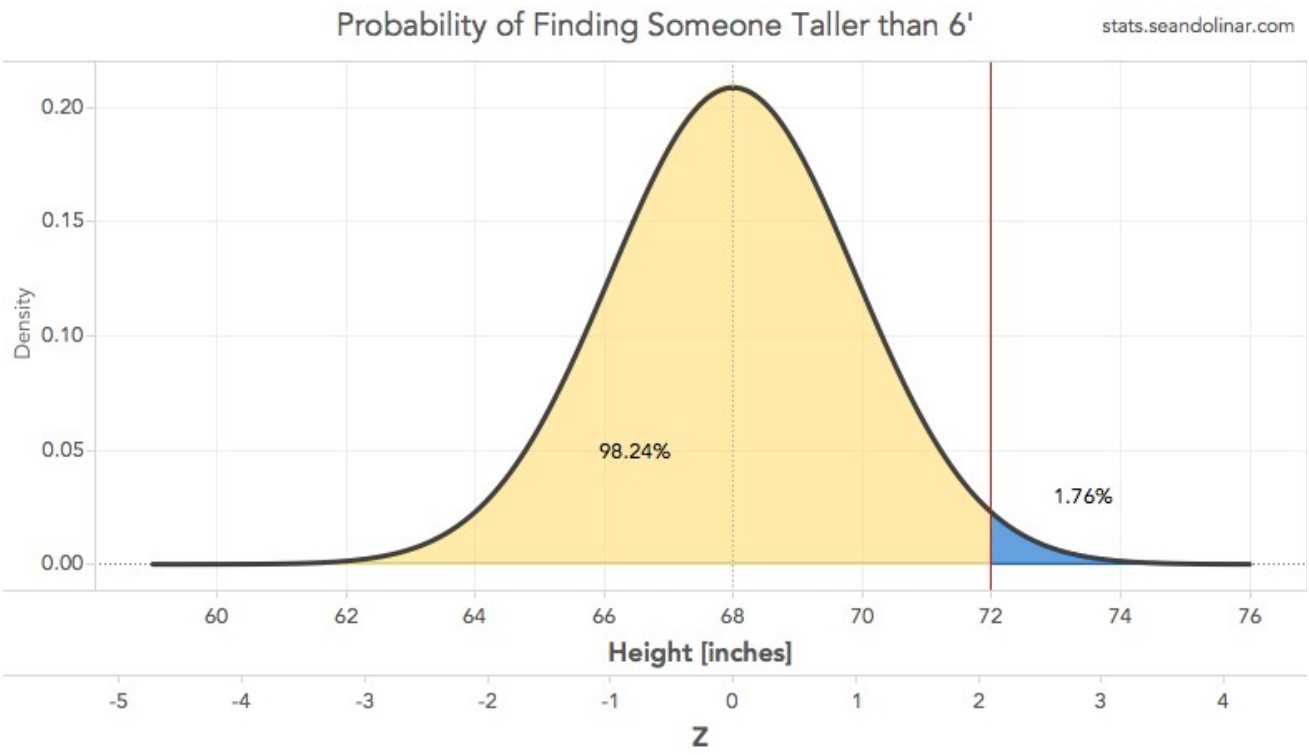
```
#DATA LOAD
data <- read.csv('Height_data.csv')
height <- data$Height

hist(height) #histogram

#POPULATION PARAMETER CALCULATIONS
pop_sd <- sd(height)*sqrt((length(height)-1)/(length(height)))
pop_mean <- mean(height)
```

Using just the population mean [μ = 67.99] and standard deviation [σ = 1.90], you can calculate the z-score for any given value of x. In this example I'll use 72 for x.

$latex z = \frac{x – \mu}{\sigma} &s=2$

```
z <- (72 - pop_mean) / pop_sd
```

This gives you a z-score of 2.107. To put this tool to use, let's use the z-score to find the probability of finding someone who is 72 inches [6-foot] tall. [Remember this data set doesn't apply to adults in the US, so these results might conflict with everyday experience.] The z-score will be used to determine the area [probability] underneath the distribution curve past the z-score value that we are interested in.
[One note is that you have to specify a range (72 to infinity) and not a single value (72). If you wanted to find people who are exactly 6-foot, not taller than 6-foot, you would have to specify the range of 71.5 to 72.5 inches. This is another problem, but this has everything to do with definite integrals intervals if you are familiar with Calc I.]



The above graph shows the area we intend to calculate. The blue area is our target, since it represents the probability of finding someone taller than 6-foot. The yellow area represents the rest of the population or everyone is is under 6-feet tall. The z-score and actual height measurements are both given underscoring the relationship between the two.

Typically in an introductory stats class, you'd use the z-score and look it up in a table and find the probability that way. R has a function 'pnorm' which will give you a more precise answer than a table in a book. ['pnorm' stands for "probability normal distribution".] Both R and typical z-score tables will return the area under the curve from -infinity to value on the graph this is represented by the yellow area. In this particular problem, we want to find the blue area. The solution to this is

an easy arithmetic function. The area under the curve is 1, so by subtracting the yellow area from 1 will give you the area [probability] for the blue area.

Yellow Area:

```
p_yellow1 <- pnorm(72, pop_mean, pop_sd)    #using x, mu, and sigma
p_yellow2 <- pnorm(z)                       #using z-score of 2.107
```

Blue Area [TARGET]:

```
p_blue1 <- 1 - p_yellow1   #using x, mu, and sigma
p_blue2 <- 1 - p_yellow2   #using z-score of 2.107
```

Both of these techniques in R will yield the same answer of 1.76%. I used both methods, to show that R has some versatility that traditional statistics tables don't have. I personally find statistics tables antiquated, since we have better ways to determine it, and the table doesn't help provide any insight over software solutions.

Z-scores are useful when relating different measurement distributions to each acting as a 'common denominator'. The z-scores are used extensively for determining area underneath the curve when using text book tables, and also can be easily used in programs such as R. Some statistical hypothesis tests are based on z-scores and the basic principles of finding the area beyond some value.

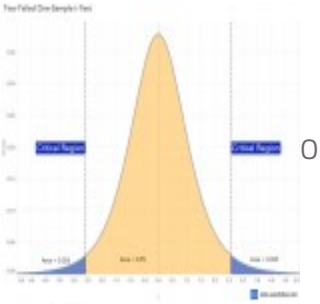
FEATURED HEIGHT NORMAL DISTRIBUTION STATISTICS Z-SCORE

PREVIOUS POST
The Most Popular Emoji Characters on Twitter

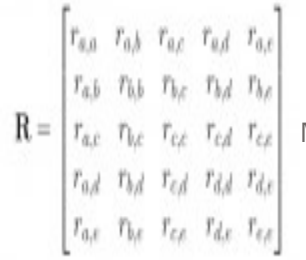
NEXT POST
One Mean Z-test [with R code]

Follow @seandolinar

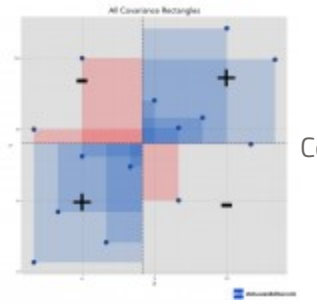
RELATED POSTS



One-Sample t-Test [With R Code]

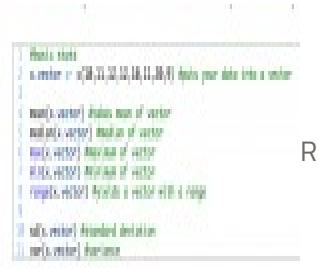


Making a Correlation Matrix in R



Covariance — Different Ways to Explain or Visualize It

D3 Visualization Basics — First Steps



R Bootcamp — A Quick Introduction