Lab 3 Tutorial

Janette Avelar & Anwesha Guha

1/25/2022

Intro

This week we will go over calculating probability from a standard normal distribution.

Note: **All material is taken from Dr. Zopluoglu's 2021 EDUC 614 materials and adapted for this year's course.**

We have adapted it to slide format using R.

Want to learn how we created R slide presentations? This tutorial *is helpful.*

Calculating Probability from a Standard Normal Distribution

Density for z-score

To compute density for a z-score, you can compute `dnorm()` from the base package. Below is the code to reproduce density calculations from Lecture 3a.

```
dnorm(x = -1)
```

```
## [1] 0.2419707
```

```
dnorm(x = 0)
```

```
## [1] 0.3989423
```

```
dnorm(x = 1)
```

```
## [1] 0.2419707
```

```
dnorm(x = 2)
```

```
## [1] 0.05399097
```

Lower-tail Probability

Lower-tail probability refers the probability of a random variable X taking the value x or less. In other words, it refers to the area under the normal curve on the left side of normal distribution.

Cumulative probability is also another term being used for lower-tail probability.

You can compute this by using the `pnorm()` function from the base package.

Examples

Suppose, you want to compute the probability of z scores being 0 or less than 0.

P(z < 0) = ?

```
pnorm(0, lower.tail = TRUE)
```

```
## [1] 0.5
```

What about probability of z score being -1 or less than -1?

P(z < -1) = ?

```
pnorm(-1, lower.tail = TRUE)
```

```
## [1] 0.1586553
```

Or probability of z score being 2 or less than 2?

$P(z < 2) = ?$

```
pnorm(2, lower.tail = TRUE)
```

```
## [1] 0.9772499
```

Upper-tail Probability

Upper-tail probability refers the probability of a random variable X taking the value x or more. In other words, it refers to the area under the normal curve on the right side of normal distribution.

You can compute this by using `pnorm()` from the base package, and setting `lower.tail = FALSE`.

Examples

Suppose you want to calculate the probability of z score being -1 or more than -1.

P(z > -1) = ?

```
pnorm(-1, lower.tail = FALSE)
```

```
## [1] 0.8413447
```

What about the probability of z score being 2 or more than 2?

P(z > 2) = ?

```
pnorm(2, lower.tail = FALSE)
```

```
## [1] 0.02275013
```

Probability between scores

If you want to compute the probability between two z-scores, then you need to get creative.

Suppose you want to compute the probability of z score being between -1 and 2.

You can do this by subtracting P(z<-1) from P(z<2).

```
pnorm(2, lower.tail = TRUE) -  pnorm(-1, lower.tail = TRUE)
```

```
## [1] 0.8185946
```

`qnorm()`

Another useful function is `qnorm()`. Consider this function as an inverse of `pnorm()`.

- `pnorm()` returns the cumulative probability given a z-score in a normal distribution
- `qnorm()` returns the z-score given the cumulative probability.

Example

Suppose you want to know that which z-score has a cumulative probability of 0.6.

```
qnorm(0.6,lower.tail=TRUE)
```

```
## [1] 0.2533471
```

Q-Q Plots

Import the data

Like in our previous tutorial where we learned how to create histograms and boxplots, in order to create a q-q plot our first step is to import the dataset we'd like to use.

```
Add <- read.csv(file   = "/Users/janetteavelar/Desktop/Add.csv",
                header = TRUE)
```

Remember your working directory goes in the first line of code. You can set it manually, or use `getwd()` to find your file path.

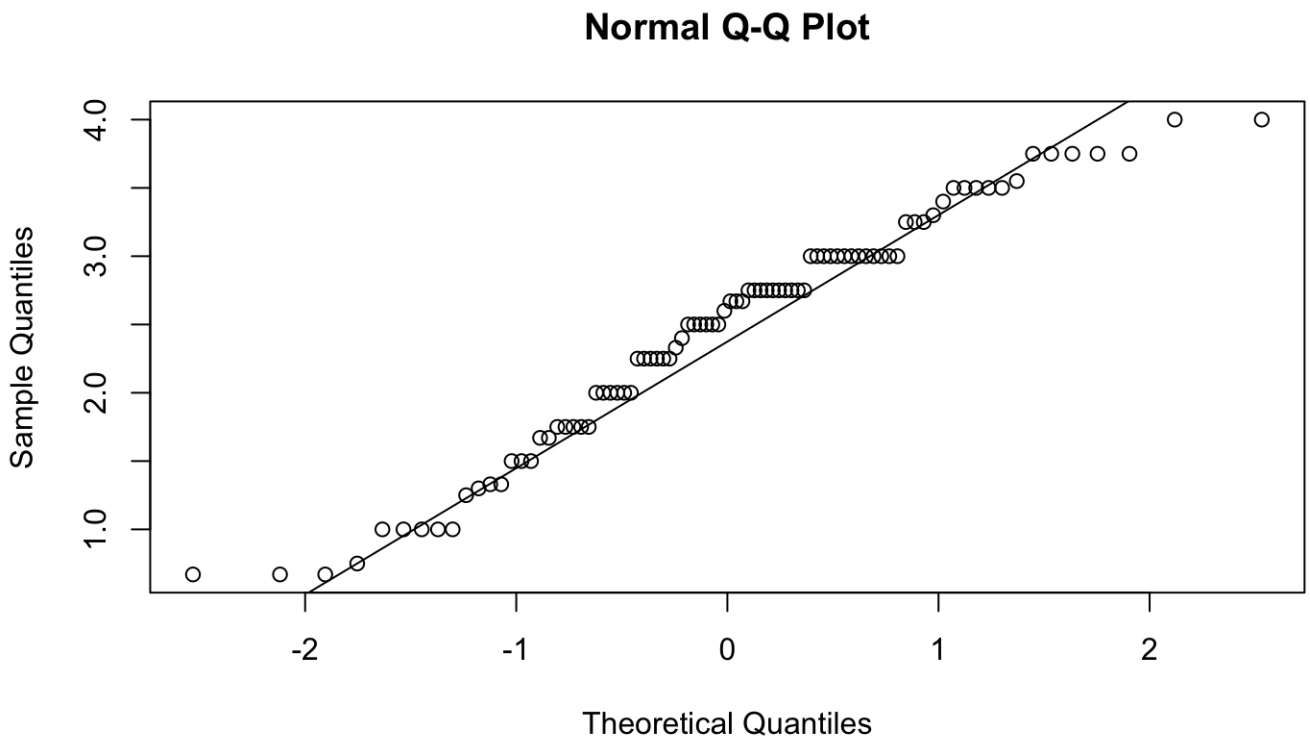
Note: If you choose to import your dataset using the point-and-click interface, be sure to mark **Yes** for Header to ensure your data loads in properly.

Create the plot

We'll now use the `qqline()` function from the base package to draw a QQ plot for any variable of choice.

Let's look at GPA first.

```
qqnorm(Add$GPA)
qqline(Add$GPA)  # This adds a diagonal line
```



Another example

Now let's try another variable.

```
qqnorm(Add$ADDSC)
qqline(Add$ADDSC)  # Add the diagonal line
```