The normal distribution

Heleine Fouda

In this lab, you'll investigate the probability distribution that is most central to statistics: the normal distribution. If you are confident that your data are nearly normal, that opens the door to many powerful statistical methods. Here we'll use the graphical tools of R to assess the normality of our data and also learn how to generate random numbers from a normal distribution.

Getting Started

Load packages

In this lab, we will explore and visualize the data using the **tidyverse** suite of packages as well as the **openintro** package.

Let's load the packages.

```
library(tidyverse)
library(openintro)
```

The data

This week you'll be working with fast food data. This data set contains data on 515 menu items from some of the most popular fast food restaurants worldwide. Let's take a quick peek at the first few rows of the data.

Either you can use `glimpse` like before, or `head` to do this.

```
library(tidyverse)
library(openintro)
data("fastfood", package='openintro')
head(fastfood)

## # A tibble: 6 × 17
##   restaurant item       calories cal_fat total_fat sat_fat trans_fat
cholesterol
##   <chr>      <chr>         <dbl>   <dbl>     <dbl>   <dbl>     <dbl>
<dbl>
## 1 Mcdonalds  Artisan G…      380      60         7       2       0
95
## 2 Mcdonalds  Single Ba…      840     410        45      17       1.5
130
## 3 Mcdonalds  Double Ba…     1130     600        67      27       3
220
## 4 Mcdonalds  Grilled B…      750     280        31      10       0.5
155
```

```
## 5 Mcdonalds  Crispy Ba…      920     410         45      12        0.5
120
## 6 Mcdonalds  Big Mac         540     250         28      10        1
80
## # ℹ 9 more variables: sodium <dbl>, total_carb <dbl>, fiber <dbl>, sugar
<dbl>,
## #   protein <dbl>, vit_a <dbl>, vit_c <dbl>, calcium <dbl>, salad <chr>
```

You'll see that for every observation there are 17 measurements, many of which are nutritional facts.

You'll be focusing on just three columns to get started: restaurant, calories, calories from fat.

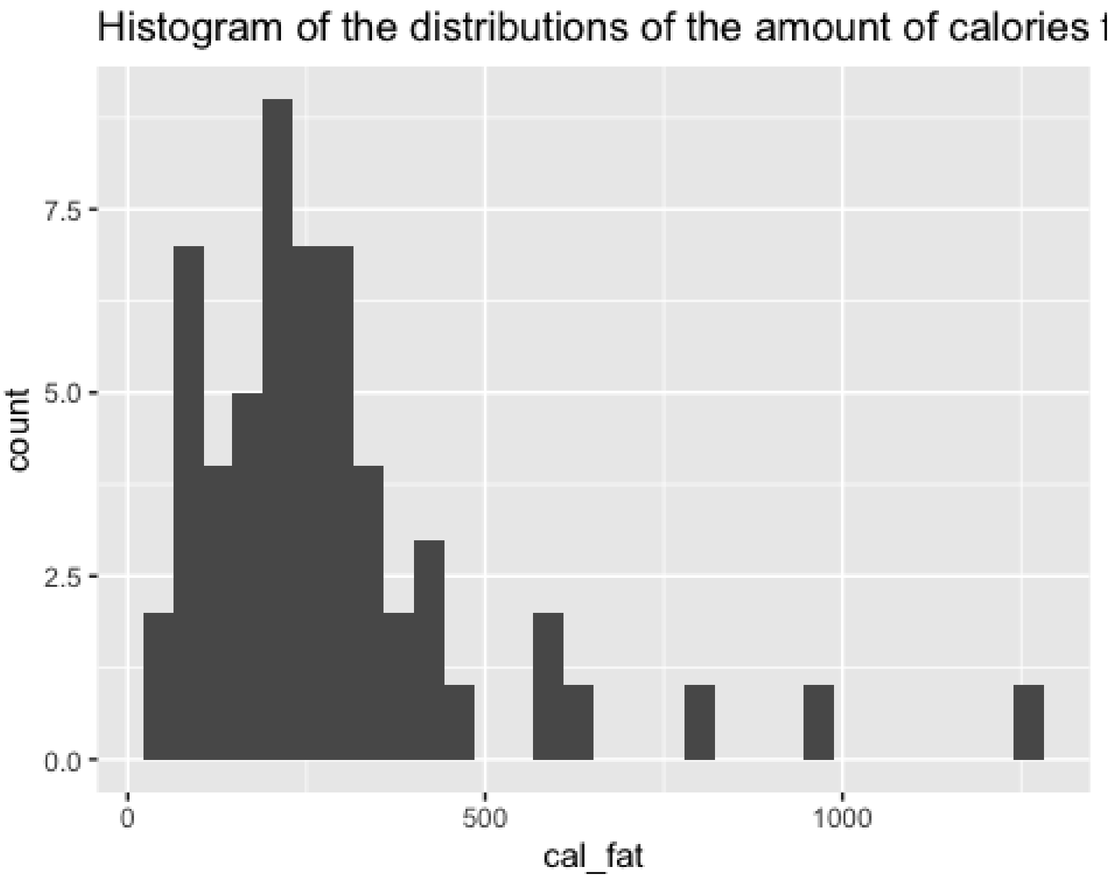
Let's first focus on just products from McDonalds and Dairy Queen.

```
mcdonalds<- fastfood %>%
  filter(restaurant == "Mcdonalds")
dairy_queen <- fastfood %>%
  filter(restaurant == "Dairy Queen")
```

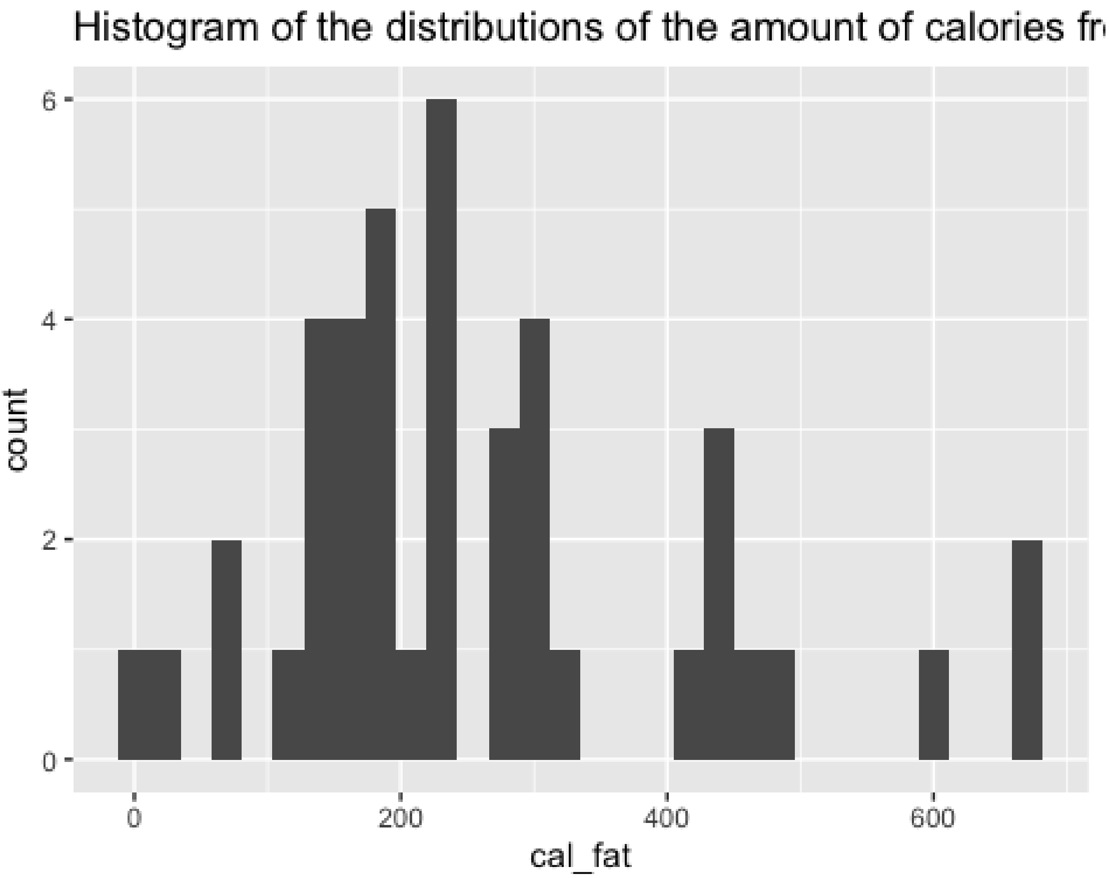
1. Make a plot (or plots) to visualize the distributions of the amount of calories from fat of the options from these two restaurants. How do their centers, shapes, and spreads compare?

Insert your answer here In both histograms, most observations are found in the center/middle and then spread decreasingly and symmetrically from that center or middle point, forming what looks like a bell curve.

```
  ggplot(data = mcdonalds, aes(x = cal_fat)) +
  geom_histogram() +
  labs(title =  "Histogram of the distributions of the amount of calories
from fat of the options from mcdonalds")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(data = dairy_queen, aes(x = cal_fat)) +
  geom_histogram() +
  labs(title =  "Histogram of the distributions of the amount of calories
from fat of the options from dairy_queen")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The normal distribution

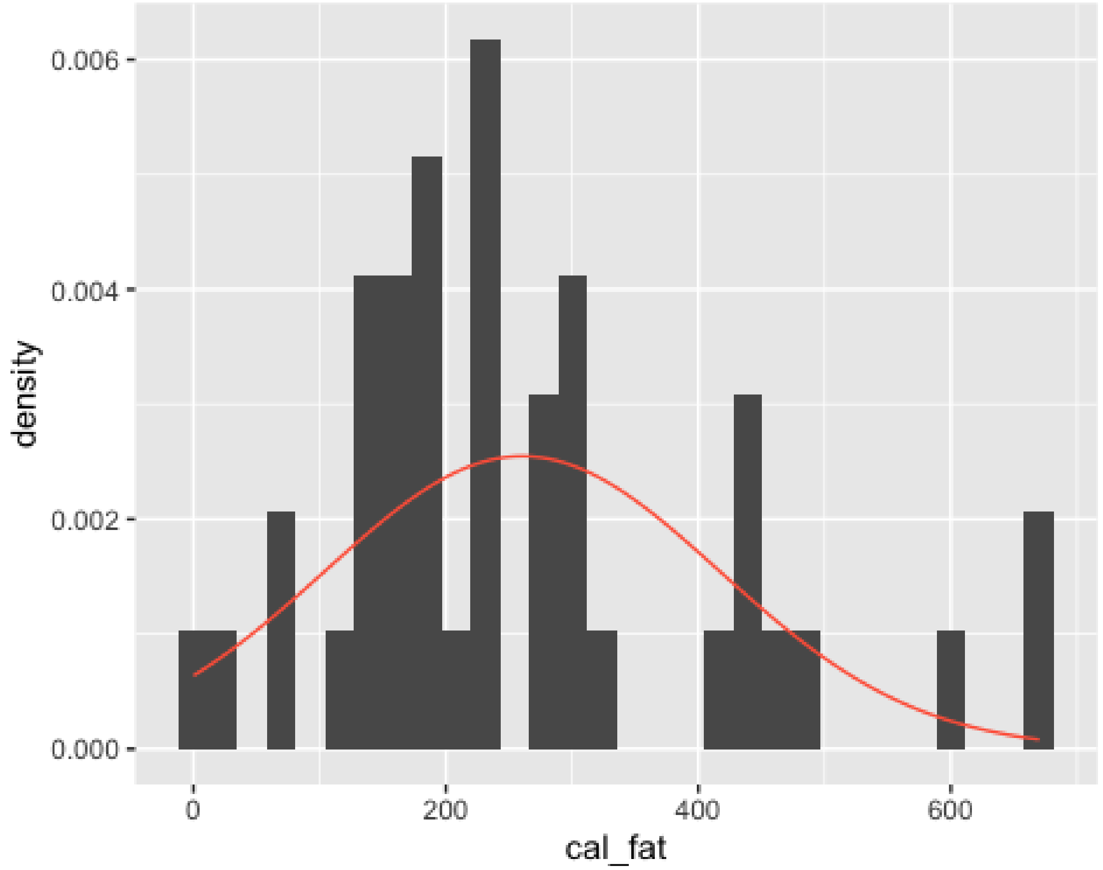
In your description of the distributions, did you use words like *bell-shaped*or *normal*? It's tempting to say so when faced with a unimodal symmetric distribution.

To see how accurate that description is, you can plot a normal distribution curve on top of a histogram to see how closely the data follow a normal distribution. This normal curve should have the same mean and standard deviation as the data. You'll be focusing on calories from fat from Dairy Queen products, so let's store them as a separate object and then calculate some statistics that will be referenced later.

```
dqmean <- mean(dairy_queen$cal_fat)
dqsd   <- sd(dairy_queen$cal_fat)
```

Next, you make a density histogram to use as the backdrop and use the `lines` function to overlay a normal probability curve. The difference between a frequency histogram and a density histogram is that while in a frequency histogram the *heights* of the bars add up to the total number of observations, in a density histogram the *areas* of the bars add up to 1. The area of each bar can be calculated as simply the height *times* the width of the bar. Using a density histogram allows us to properly overlay a normal distribution curve over the histogram since the curve is a normal probability density function that also has area under the curve of 1. Frequency and density histograms both display the same exact shape; they only differ in their y-axis. You can verify this by comparing the frequency histogram you constructed earlier and the density histogram created by the commands below.

```
ggplot(data = dairy_queen, aes(x = cal_fat)) +
        geom_blank() +
        geom_histogram(aes(y = ..density..)) +
        stat_function(fun = dnorm, args = c(mean = dqmean, sd = dqsd), col =
"tomat  o")

## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2
3.4.0.
## ℹ Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



After initializing a blank plot with `geom_blank()`, the `ggplot2` package (within the `tidyverse`) allows us to add additional layers. The first layer is a density histogram. The second layer is a statistical function – the density of the normal curve, `dnorm`. We specify that we want the curve to have the same mean and standard deviation as the column of fat calories. The argument col simply sets the color for the line to be drawn. If we left it out, the line would be drawn in black.

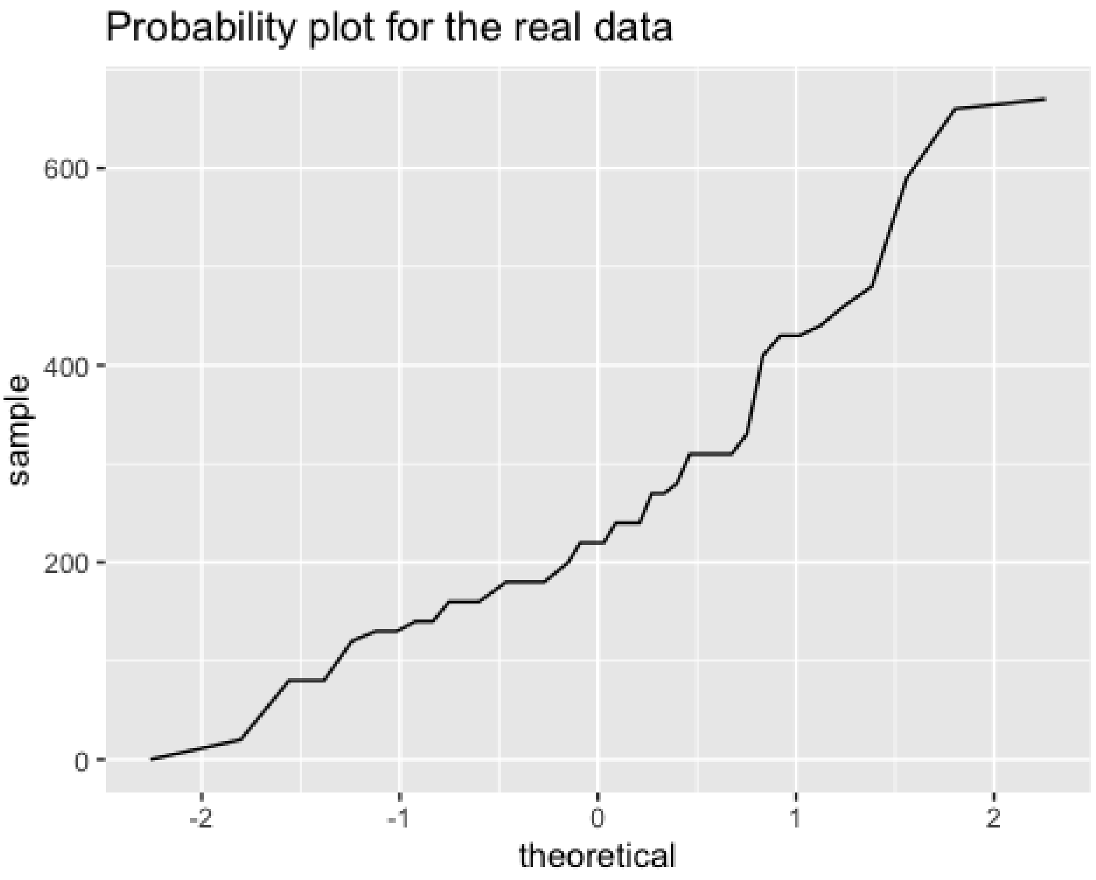
2. Based on the this plot, does it appear that the data follow a nearly normal distribution?

Yes, based on the above plot, the data appear to follow a nearly normal distribution as most of the observations (approximately 68 per cent of them)fall within 1 standard deviation above and below the mean.**

Evaluating the normal distribution

Eyeballing the shape of the histogram is one way to determine if the data appear to be nearly normally distributed, but it can be frustrating to decide just how close the histogram is to the curve. An alternative approach involves constructing a normal probability plot, also called a normal Q-Q plot for "quantile-quantile".

```
ggplot(data = dairy_queen, aes(sample = cal_fat)) +
  geom_line(stat = "qq") +
labs(title = "Probability plot for the real data")
```



This time, you can use the `geom_line()` layer, while specifying that you will be creating a Q-Q plot with the `stat` argument. It's important to note that here, instead of using `x` instead `aes()`, you need to use `sample`.

The x-axis values correspond to the quantiles of a theoretically normal curve with mean 0 and standard deviation 1 (i.e., the standard normal distribution). The y-axis values correspond to the quantiles of the original unstandardized sample data. However, even if we were to standardize the sample data values, the Q-Q plot would look identical. A data set that is nearly normal will result in a probability plot where the points closely follow a diagonal line. Any deviations from normality leads to deviations of these points from that line.

The plot for Dairy Queen's calories from fat shows points that tend to follow the line but with some errant points towards the upper tail. You're left with the same problem that we encountered with the histogram above: how close is close enough?

A useful way to address this question is to rephrase it as: what do probability plots look like for data that I *know* came from a normal distribution? We can answer this by simulating data from a normal distribution using `rnorm`.

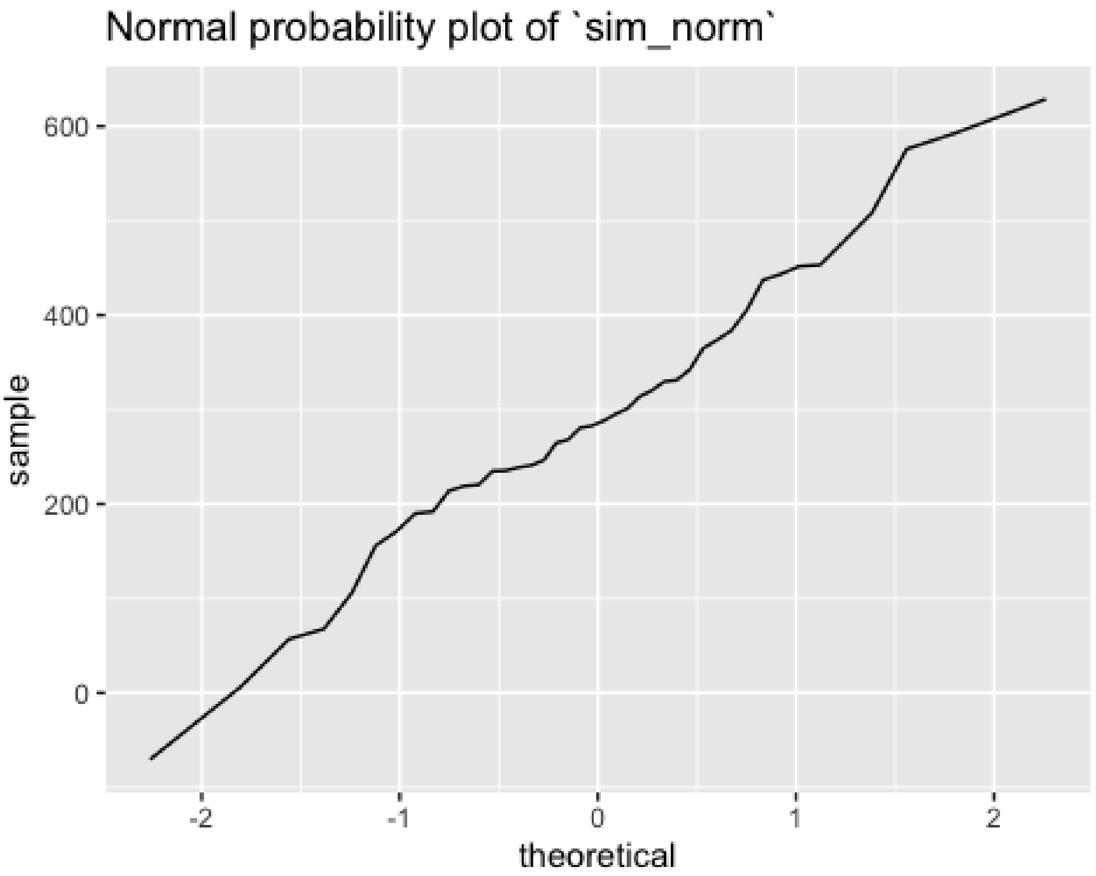
```
sim_norm <- rnorm(n = nrow(dairy_queen), mean = dqmean, sd = dqsd)
```

The first argument indicates how many numbers you'd like to generate, which we specify to be the same number of menu items in the `dairy_queen` data set using the `nrow()` function. The last two arguments determine the mean and standard deviation of the normal distribution from which the simulated sample will be generated. You can take a look at the shape of our simulated data set, `sim_norm`, as well as its normal probability plot.

3. Make a normal probability plot of `sim_norm`. Do all of the points fall on the line? How does this plot compare to the probability plot for the real data? (Since `sim_norm` is not a data frame, it can be put directly into the `sample` argument and the `data` argument can be dropped.)

Insert your answer here Compared to the probability plot for the real data, a normal probability plot of `sim_norm` (see below) appears to have an almost flawless exponential linearity.The probability plot for the real data however presents (despite its overall linearity) areas where some observations fall far away from the line.

```
ggplot(data = NULL, aes(sample = sim_norm)) +
  geom_line(stat = "qq") +
labs(title = "Normal probability plot of `sim_norm`")
```



Even better than comparing the original plot to a single plot generated from a normal distribution is to compare it to many more plots using the following function. It shows the Q-Q plot corresponding to the original data in the top left corner, and the Q-Q plots of 8 different simulated normal data. It may be helpful to click the zoom button in the plot window.

```
qqnormsim(sample = cal_fat, data = dairy_queen) +
  labs(title = "Normal probability plot simulation - Dairy Queen")
```

-1.png)

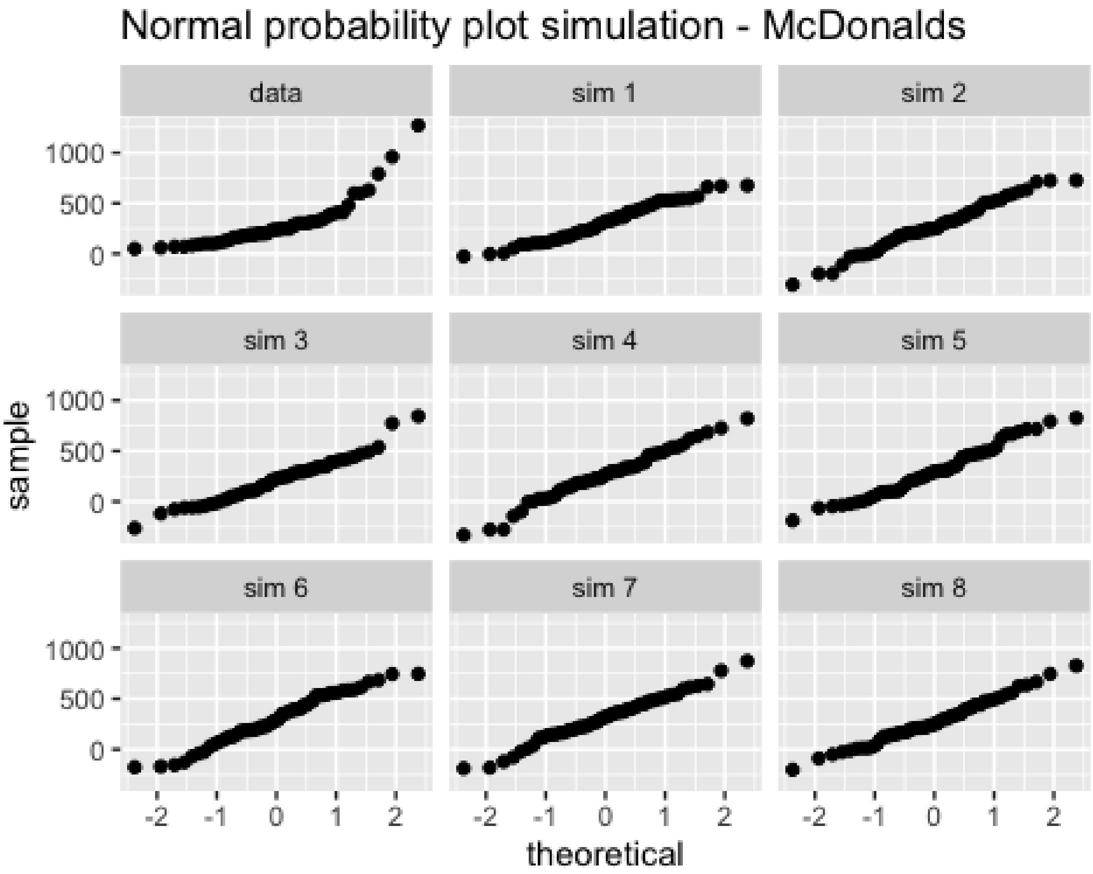
4. Does the normal probability plot for the calories from fat look similar to the plots created for the simulated data? That is, do the plots provide evidence that the calories are nearly normal?

** Yes, the normal probability plot for the calories from fat look similar to the plots created for the simulated data, and especially **sim 7** It is also a normal distribution in accordance to the section of the course on Normal Q-Q Plot that states that the existence of a linear relationship in the plot is an indicator of a near normal distribution.**

5. Using the same technique, determine whether or not the calories from McDonald's menu appear to come from a normal distribution.

Insert your answer here

```
qqnormsim(sample = cal_fat, data = mcdonalds) +
  labs(title = "Normal probability plot simulation - McDonalds")
```



In this scenario, the normal probability plot for the calories from fat does not look similar to the plots created for the simulated data. Nonetheless, all the 7 simulation plots provide evidence that the calories from mcdonals are nearly normal, as there is a linear relationship in each of the 7 simulation plots in the shape of a diagonal line.

Normal probabilities

Okay, so now you have a slew of tools to judge whether or not a variable is normally distributed. Why should you care?

It turns out that statisticians know a lot about the normal distribution. Once you decide that a random variable is approximately normal, you can answer all sorts of questions about that variable related to probability. Take, for example, the question of, “What is the probability that a randomly chosen Dairy Queen product has more than 600 calories from fat?”

If we assume that the calories from fat from Dairy Queen’s menu are normally distributed (a very close approximation is also okay), we can find this probability by calculating a Z score and consulting a Z table (also called a normal probability table). In R, this is done in one step with the function `pnorm()`.

```
1 - pnorm(q = 600, mean = dqmean, sd = dqsd)

## [1] 0.01501523
```

Note that the function `pnorm()` gives the area under the normal curve below a given value, q, with a given mean and standard deviation. Since we're interested in the probability that a Dairy Queen item has more than 600 calories from fat, we have to take one minus that probability.

Assuming a normal distribution has allowed us to calculate a theoretical probability. If we want to calculate the probability empirically, we simply need to determine how many observations fall above 600 then divide this number by the total sample size.

```
dairy_queen %>%
  filter(cal_fat > 600) %>%
  summarise(percent = n() / nrow(dairy_queen))

## # A tibble: 1 × 1
##   percent
##     <dbl>
## 1  0.0476
```

Although the probabilities are not exactly the same, they are reasonably close. The closer that your distribution is to being normal, the more accurate the theoretical probabilities will be.

6. Write out two probability questions that you would like to answer about any of the restaurants in this dataset. Calculate those probabilities using both the theoretical normal distribution as well as the empirical distribution (four probabilities in all). Which one had a closer agreement between the two methods?

Insert your answer here 1. What is the probability that a cheeseburger from Mcdonalds has less than 300 calories from fat?"

Theoretical probability:

```
# Theoretical Probability
mcdonalds|>
  filter(item == "Cheeseburger")

## # A tibble: 1 × 17
##   restaurant item       calories cal_fat total_fat sat_fat trans_fat
cholesterol
##   <chr>      <chr>         <dbl>   <dbl>     <dbl>   <dbl>     <dbl>
<dbl>
## 1 Mcdonalds  Cheesebur…      300     100        12       5       0.5
40
## # ℹ 9 more variables: sodium <dbl>, total_carb <dbl>, fiber <dbl>, sugar
```

```
<dbl>,
## #   protein <dbl>, vit_a <dbl>, vit_c <dbl>, calcium <dbl>, salad <chr>

c_mean <- mean(mcdonalds$cal_fat)
c_sd <- sd(mcdonalds$cal_fat)

1 - pnorm(q = 300, mean = c_mean, sd = c_sd)

## [1] 0.4740374
```

Empirical probaility:

```
# Empirical distribution
mcdonalds|>
  filter(cal_fat < 300) |>
  summarise(percent = n()/nrow(mcdonalds))

## # A tibble: 1 × 1
##   percent
##     <dbl>
## 1   0.632
```

2. What is the probability that a random meal at McDonalds contains over 1000g of sodium?

Theoretical probability:

```
# Theoretical Probability
mc_mean <- mean(mcdonalds$sodium)
mc_sd <- sd(mcdonalds$sodium)

1 - pnorm(q = 1000 , mean = mc_mean, sd = mc_sd)

## [1] 0.6637094
```

Empirical probability

```
# Empirical distribution
mcdonalds|>
  filter(sodium >1000) |>
  summarise(percent = n()/nrow(mcdonalds))

## # A tibble: 1 × 1
##   percent
##     <dbl>
## 1   0.649
```

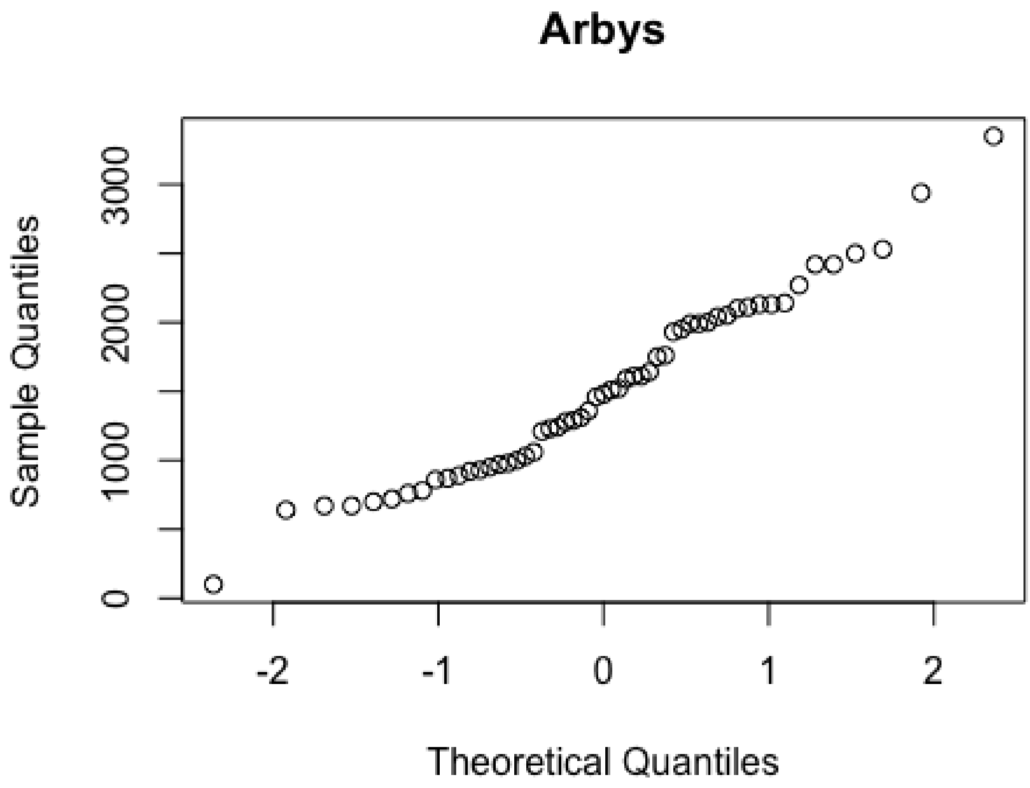
More Practice

7. Now let’s consider some of the other variables in the dataset. Out of all the different restaurants, which ones’ distribution is the closest to normal for sodium?

Insert your answer here Based on the plots below, Burger King,Taco Bell and Arbys' distributions show observations that form a near perfect diagonal line. One can then conclude that these three distributions are the closest to normal distribution for sodium.

ARBYS Sodium Plot

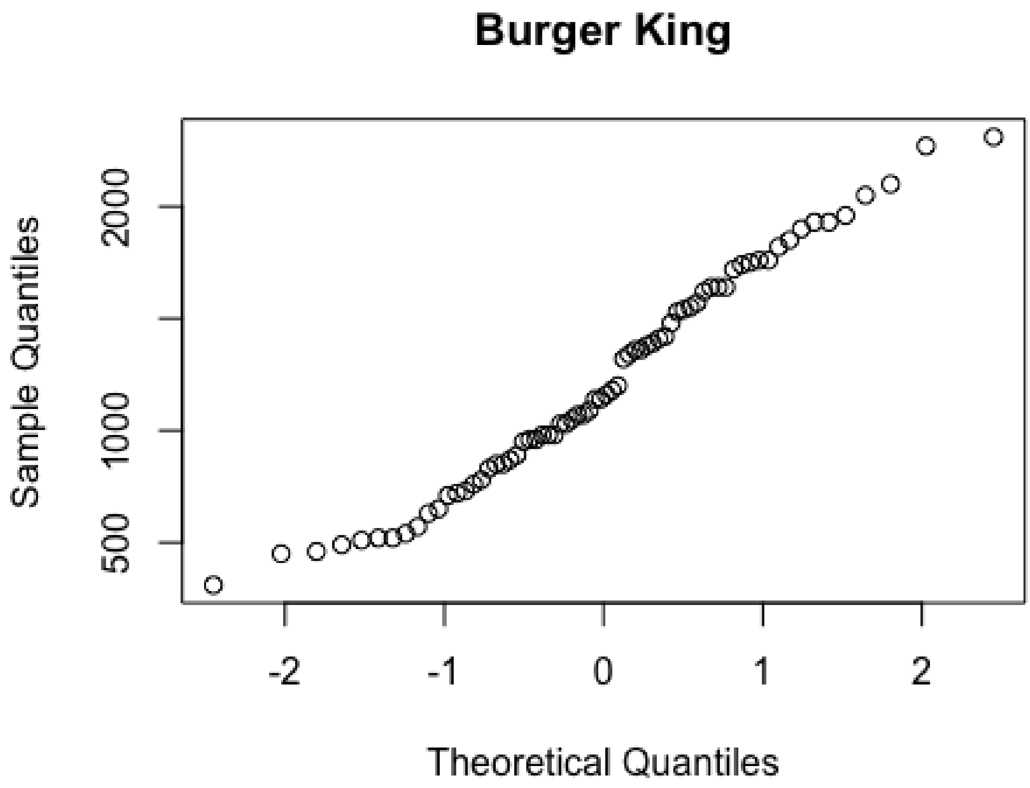
```
arbys<- fastfood|>
  filter(restaurant == "Arbys")
qqnorm(arbys$sodium, main = "Arbys")
```



```
labs(title = "Theoretical sodium distribution at Arbys")

## $title
## [1] "Theoretical sodium distribution at Arbys"
##
## attr(,"class")
## [1] "labels"
```

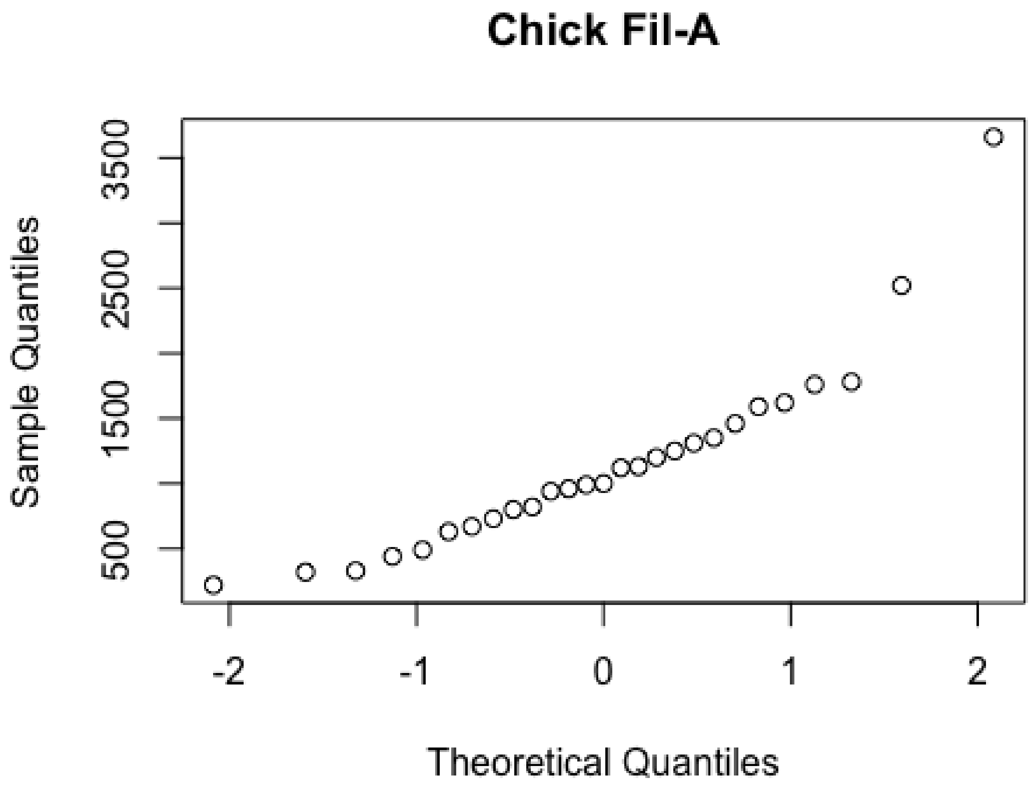
BURGER KING Sodium Plot

```
burger_king <- fastfood |>
  filter(restaurant == "Burger King")

qqnorm(burger_king$sodium, main = "Burger King")
```



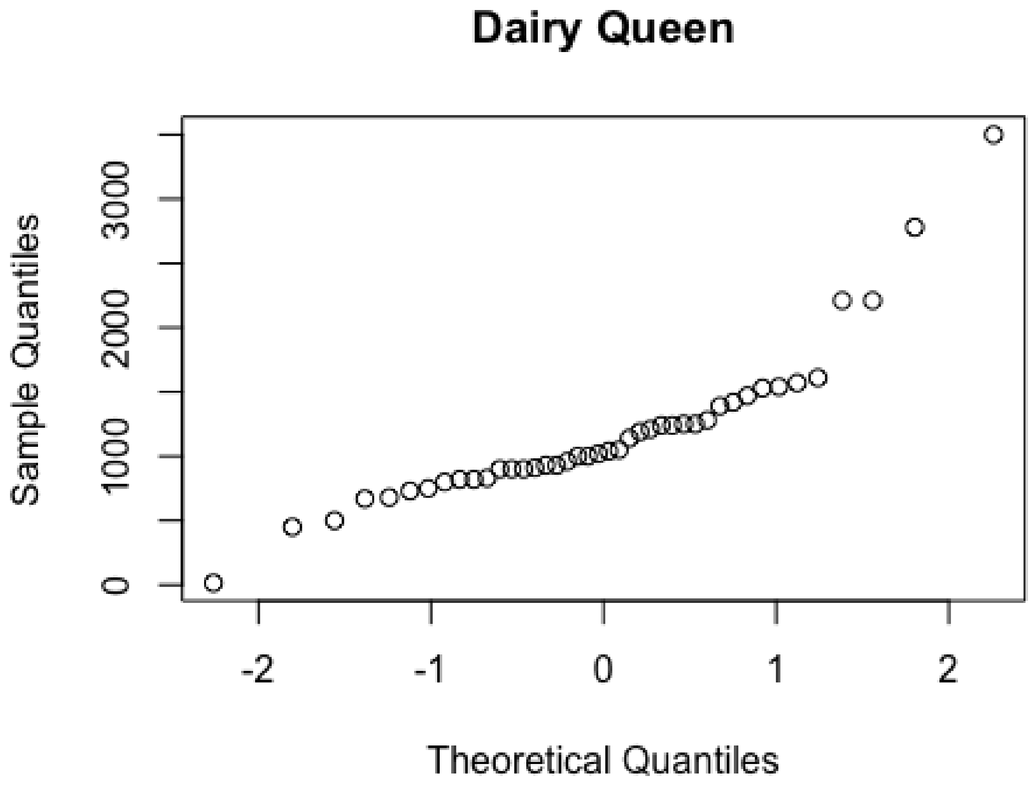
CHICK FIL A Sodium Plot

```
chick_fil_a <- fastfood|>
  filter(restaurant == "Chick Fil-A")

qqnorm(chick_fil_a $sodium, main = "Chick Fil-A")
```



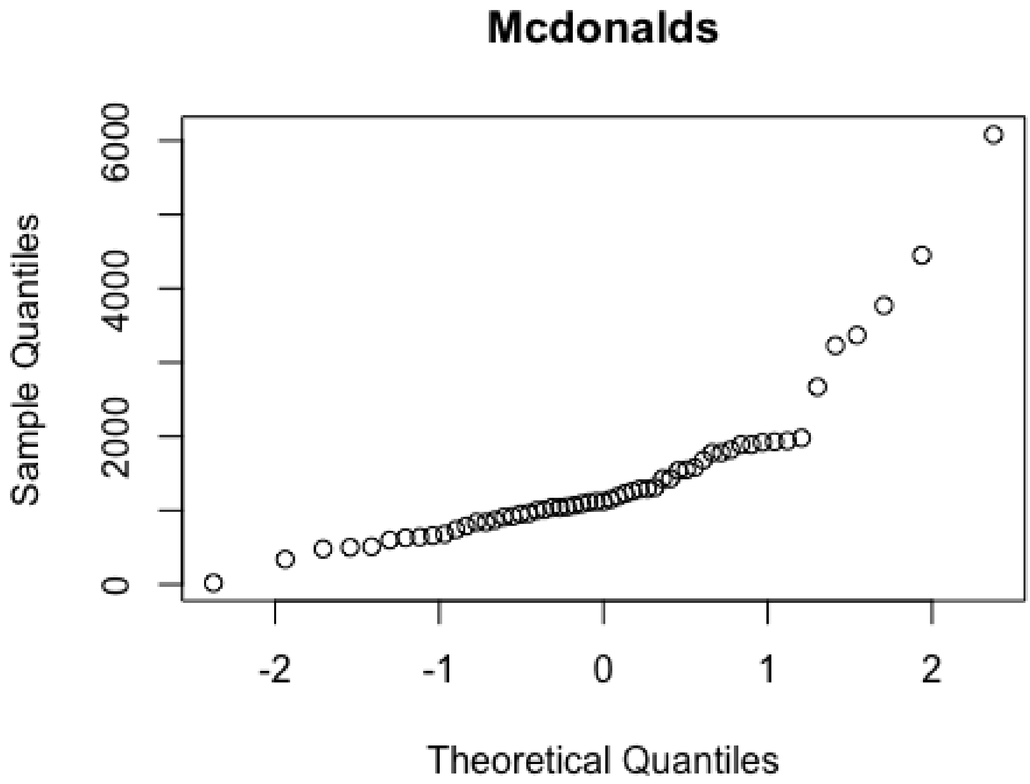
DAIRY QUEEN Sodium Plot

```
 day_q <- fastfood|>
  filter(restaurant == "Dairy Queen")
qqnorm(day_q$sodium, main = "Dairy Queen")
```



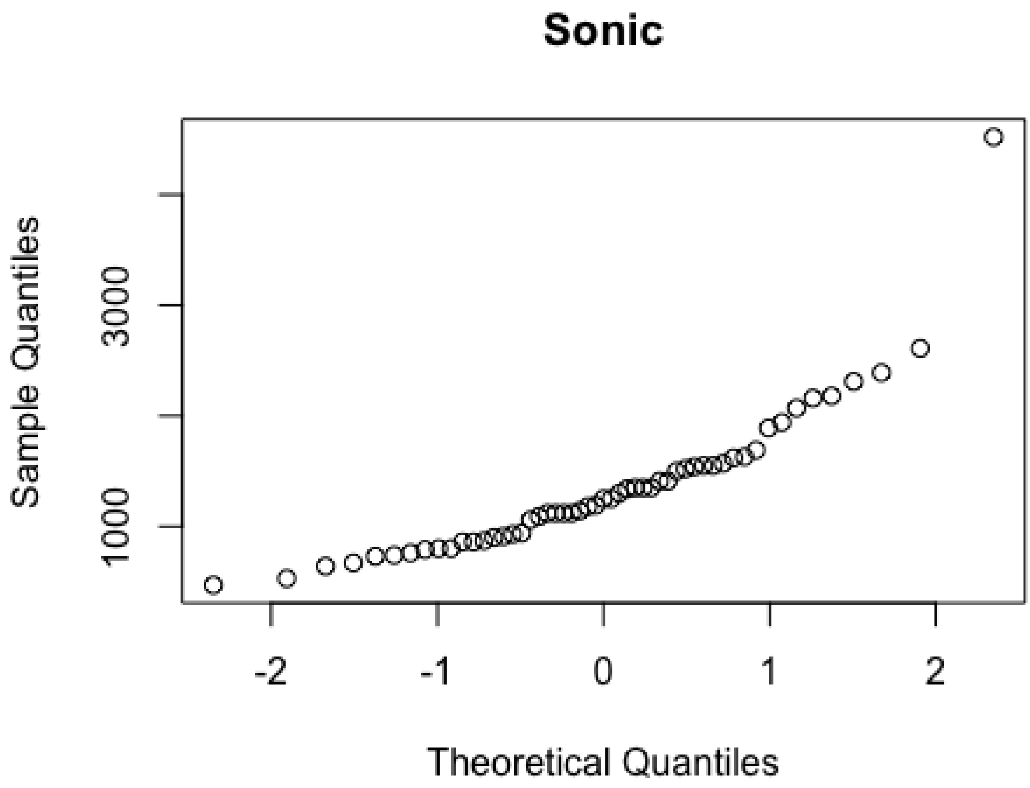
MCDONALDS Sodium Plot

```
mc_d<- fastfood|>
  filter(restaurant == "Mcdonalds")
qqnorm(mc_d$sodium, main = "Mcdonalds")
```



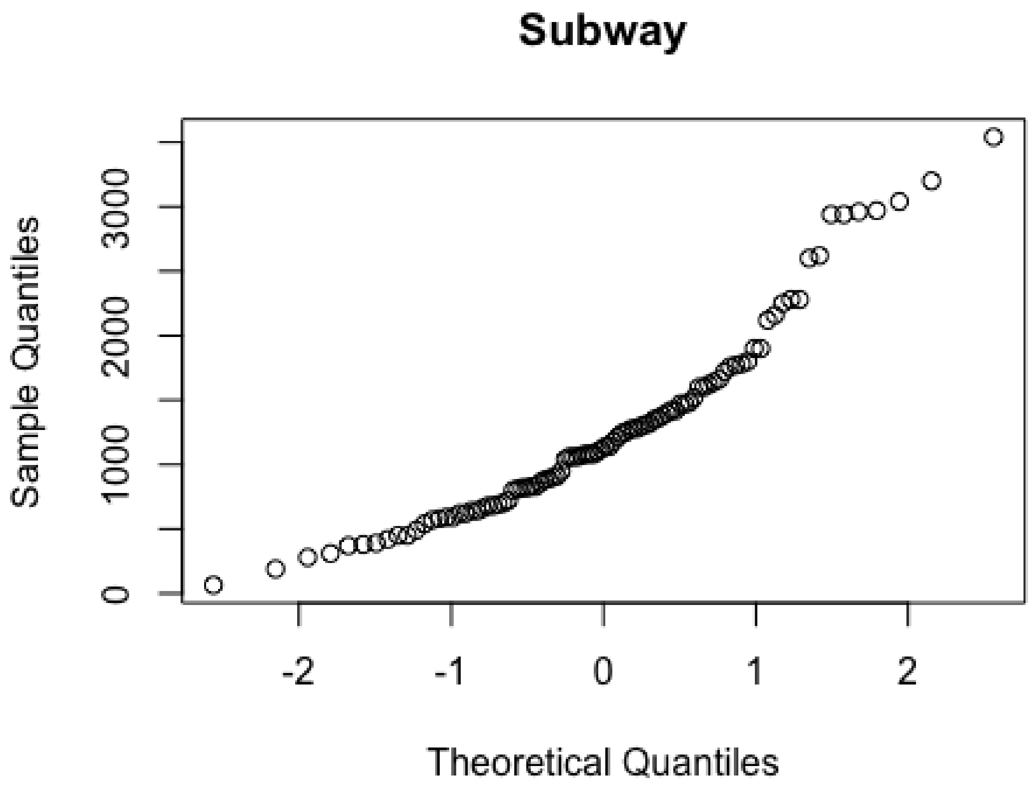
SONIC Sodium Plot

```
sonic <- fastfood|>
  filter(restaurant == "Sonic")
qqnorm(sonic$sodium, main = "Sonic")
```



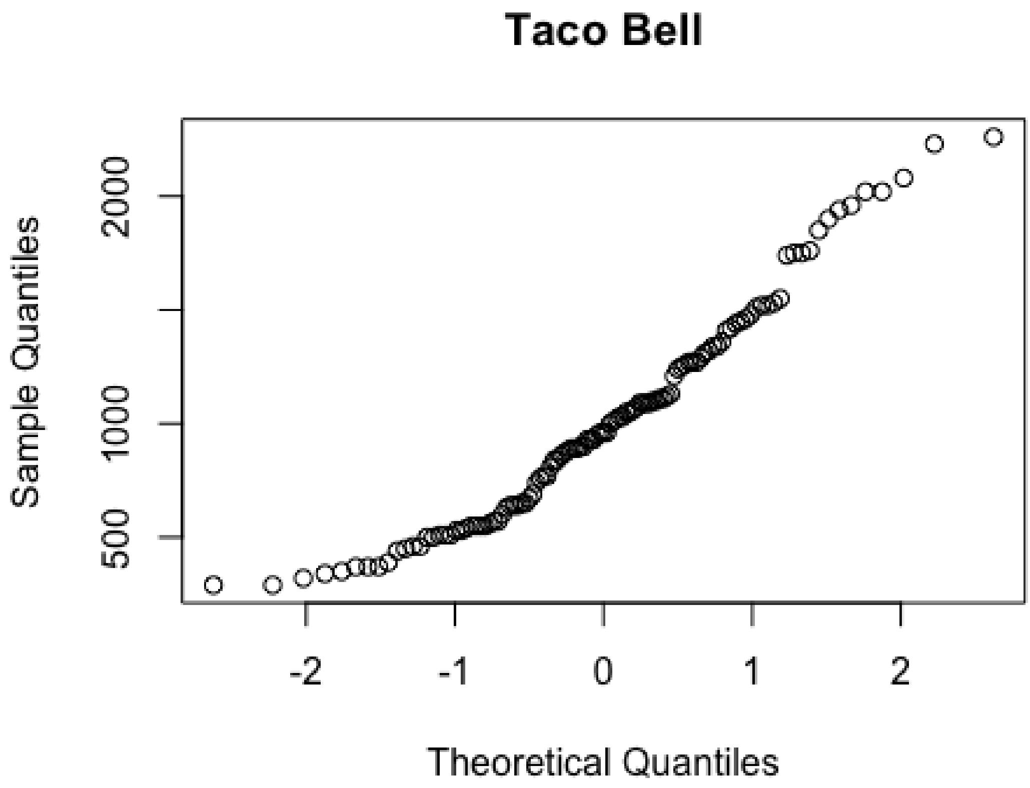
SUBWAY Sodium Plot

```
sbw <- fastfood|>
  filter(restaurant == "Subway")
qqnorm(sbw$sodium, main = "Subway")
```



TACO BELL Sodium Plot

```
taco_b <- fastfood|>
  filter(restaurant == "Taco Bell")
qqnorm(taco_b$sodium, main = "Taco Bell")
```



8. Note that some of the normal probability plots for sodium distributions seem to have a stepwise pattern. why do you think this might be the case?

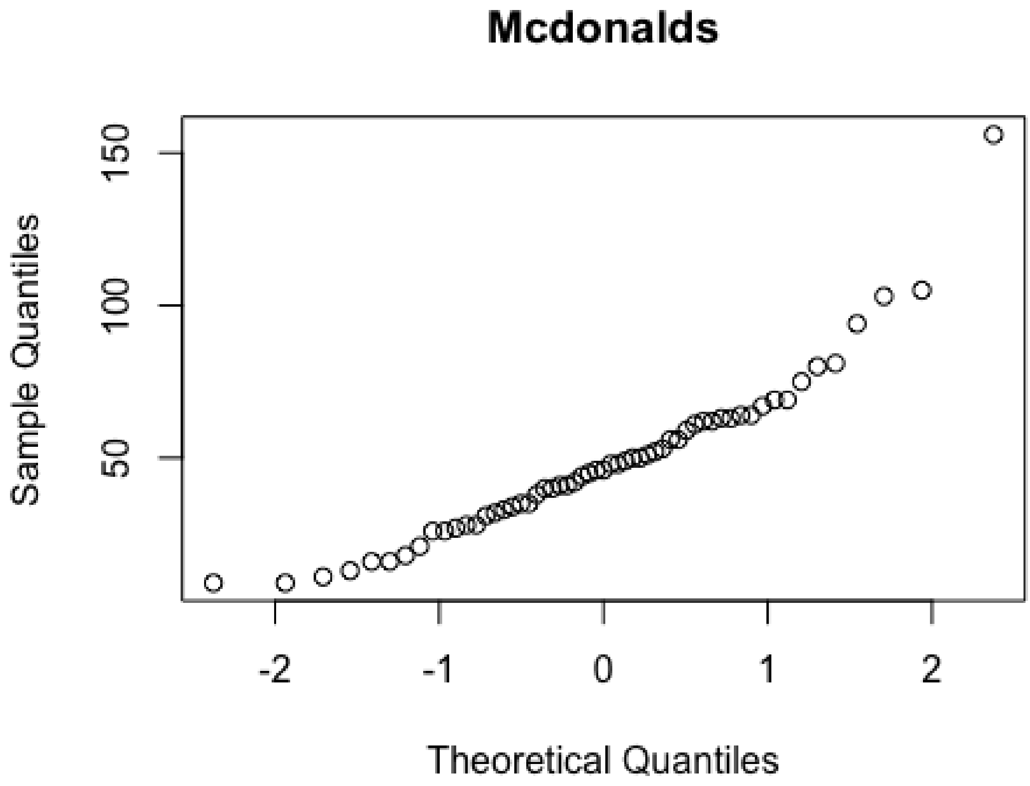
The stepwise pattern in some of the distributions may be resulting from the fact fast food chains often offer a large array of products, containing various levels of sodium.

9. As you can see, normal probability plots can be used both to assess normality and visualize skewness. Make a normal probability plot for the total carbohydrates from a restaurant of your choice. Based on this normal probability plot, is this variable left skewed, symmetric, or right skewed? Use a histogram to confirm your findings.

Insert your answer here

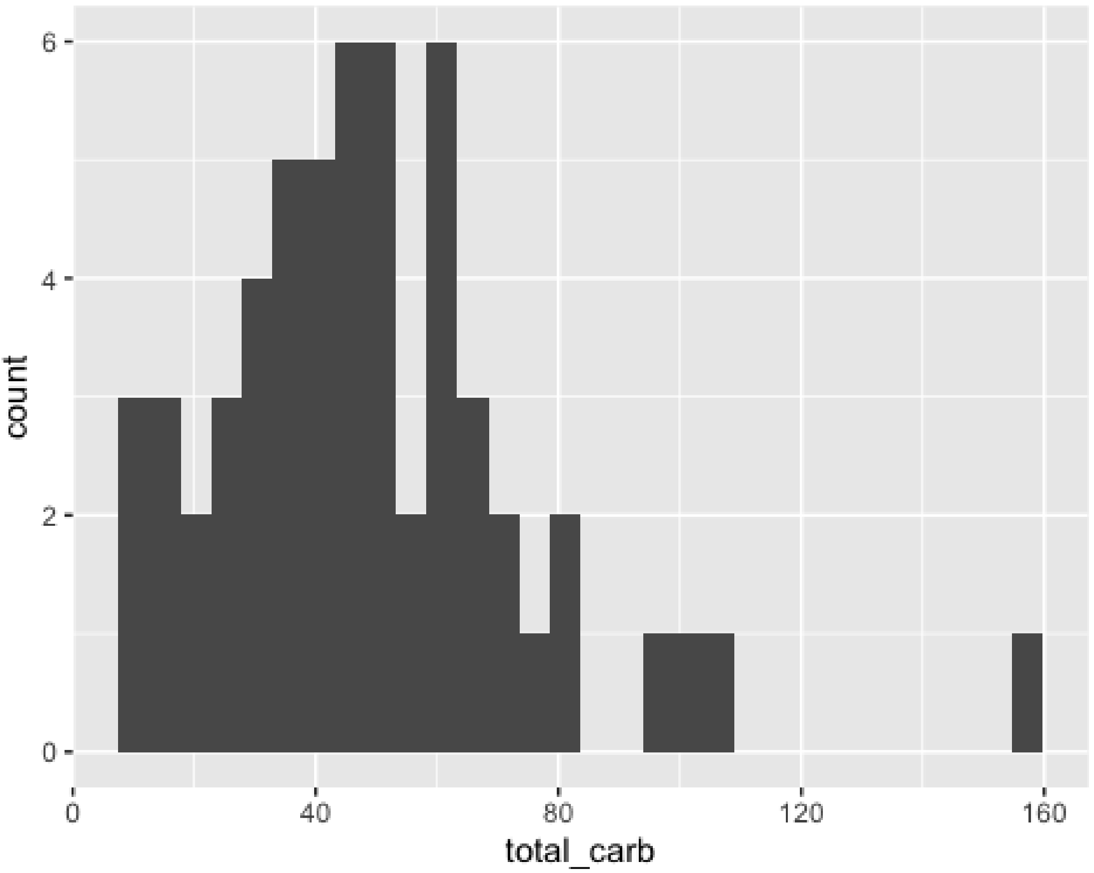
Based on the normal probability plot below, the variable **total_carb** is right skewed and the histogram confirms this with data being concentrated on the left with a tail running intermittently to the right.

```
mc_d<- fastfood|>
  filter(restaurant == "Mcdonalds")
qqnorm(mc_d$total_carb, main = "Mcdonalds")
```



```
ggplot(data = mc_d, aes(x = total_carb)) +
geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



count
6
4
2
0
0
40
80
120
160
total_carb