Data 606 Lab 4: The normal distribution

Kayleah Griffen

2/28/2023

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 x 17
##   restaur~1 item  calor~2 cal_fat total~3 sat_fat trans~4 chole~5 sodium total~6
##   <chr>     <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>  <dbl>   <dbl>
## 1 Mcdonalds Arti~     380      60       7       2       0      95   1110      44
## 2 Mcdonalds Sing~     840     410      45      17     1.5     130   1580      62
## 3 Mcdonalds Doub~    1130     600      67      27       3     220   1920      63
## 4 Mcdonalds Gril~     750     280      31      10     0.5     155   1940      62
## 5 Mcdonalds Cris~     920     410      45      12     0.5     120   1980      81
## 6 Mcdonalds Big ~     540     250      28      10       1      80    950      46
## # ... with 7 more variables: fiber <dbl>, sugar <dbl>, protein <dbl>,
```

```
## #   vit_a <dbl>, vit_c <dbl>, calcium <dbl>, salad <chr>, and abbreviated
## #   variable names 1: restaurant, 2: calories, 3: total_fat, 4: trans_fat,
## #   5: cholesterol, 6: total_carb
```

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?

```
mcdonalds |>
  ggplot(aes(x=cal_fat)) +
  geom_histogram(binwidth=30)
```
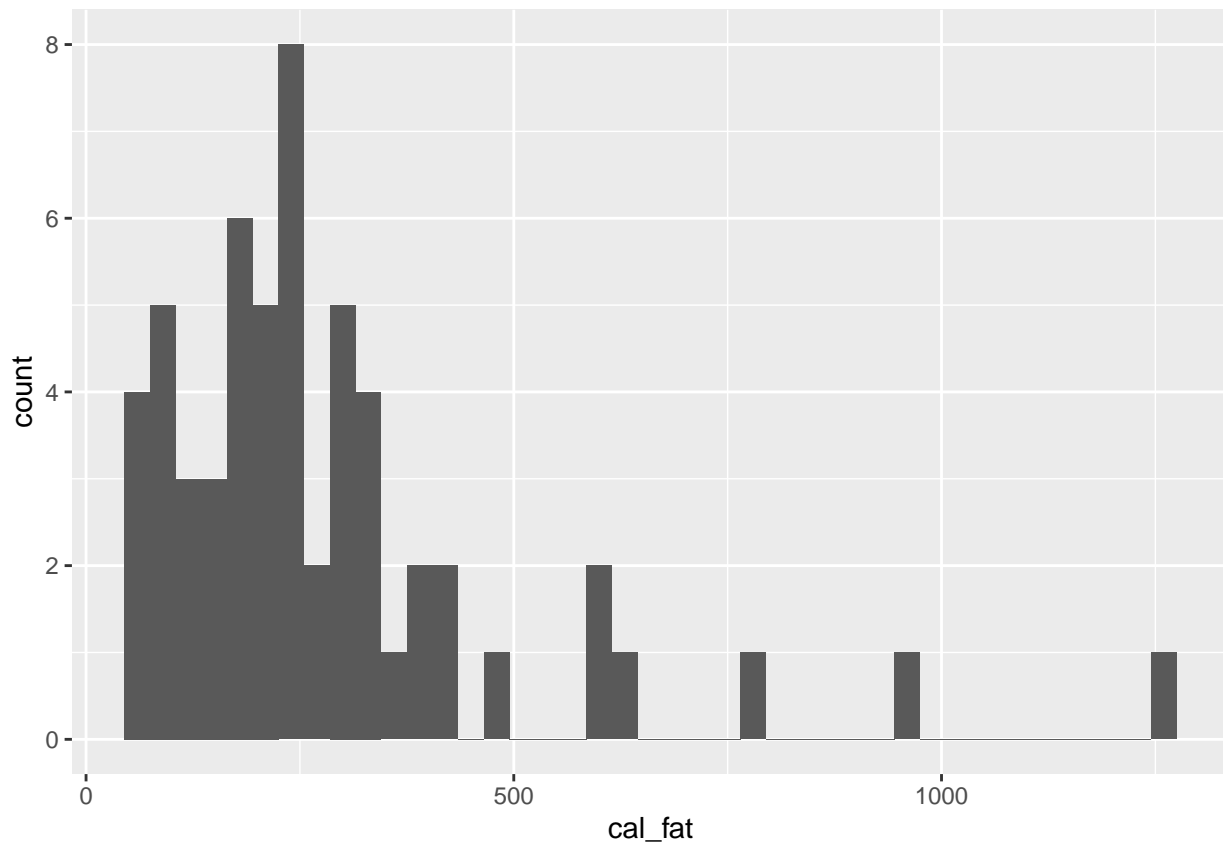


```
dairy_queen |>
  ggplot(aes(x=cal_fat)) +
  geom_histogram(binwidth=30)
```
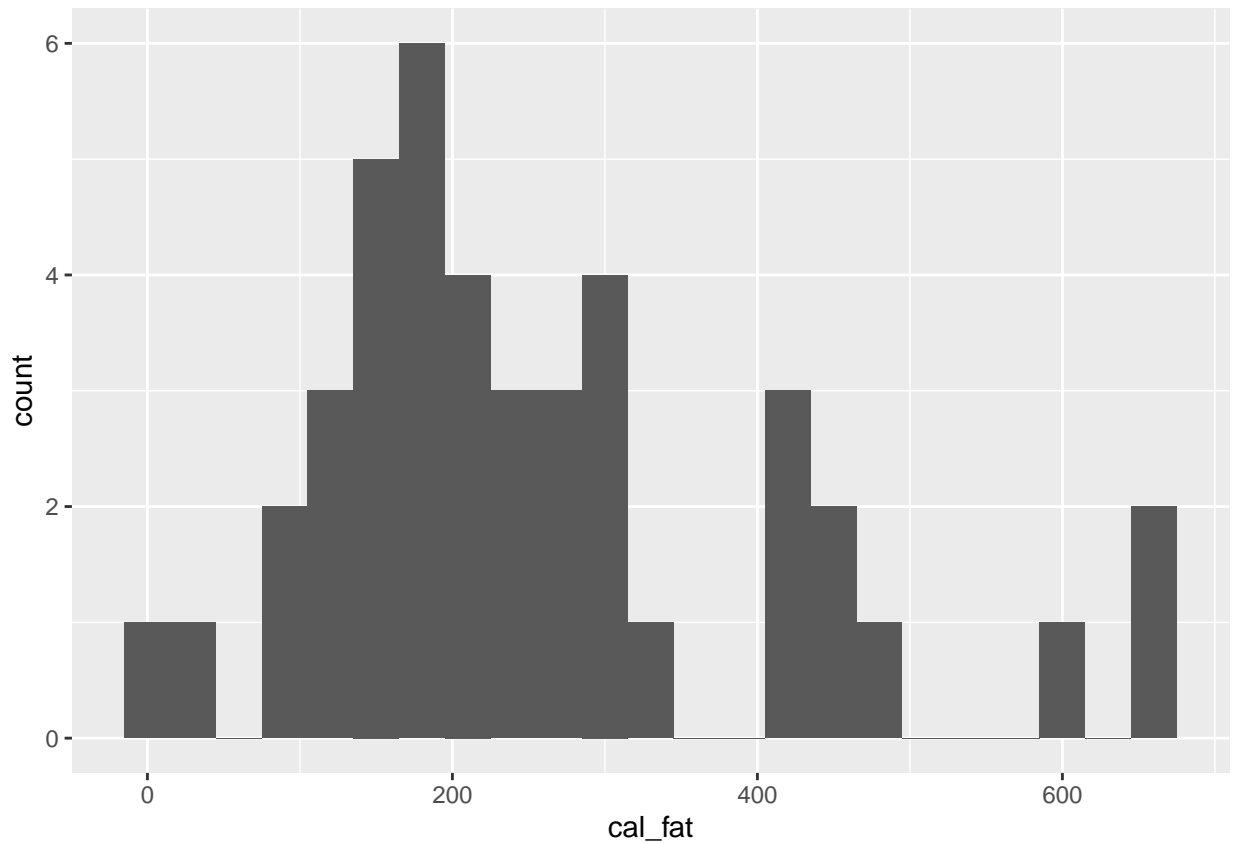
```
mcdonalds |>
  summarise(meam_m = mean(cal_fat),
            stdev_m = sd(cal_fat))
```

```
## # A tibble: 1 x 2
##   meam_m stdev_m
##    <dbl>   <dbl>
## 1   286.    221.
```

```
dairy_queen |>
  summarise(m = mean(cal_fat),
            stdev = sd(cal_fat))
```

```
## # A tibble: 1 x 2
##       m stdev
##   <dbl> <dbl>
## 1  260.  156.
```

I made two plots, one histogram for each calories from fat. The mcdonalds data follows mostly a normal distribution with a slight skew to the right - its mean is 286 and its standard deviation is 221. Dairy queen follows approximately a normal distribution with a mean of 260 and standard deviation of 156. The center of the mcdonalds distribution is slightly greater than the center of the dairy queen, mcdonalds is also more spread than the dairy queen.

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 = "tomato")
```
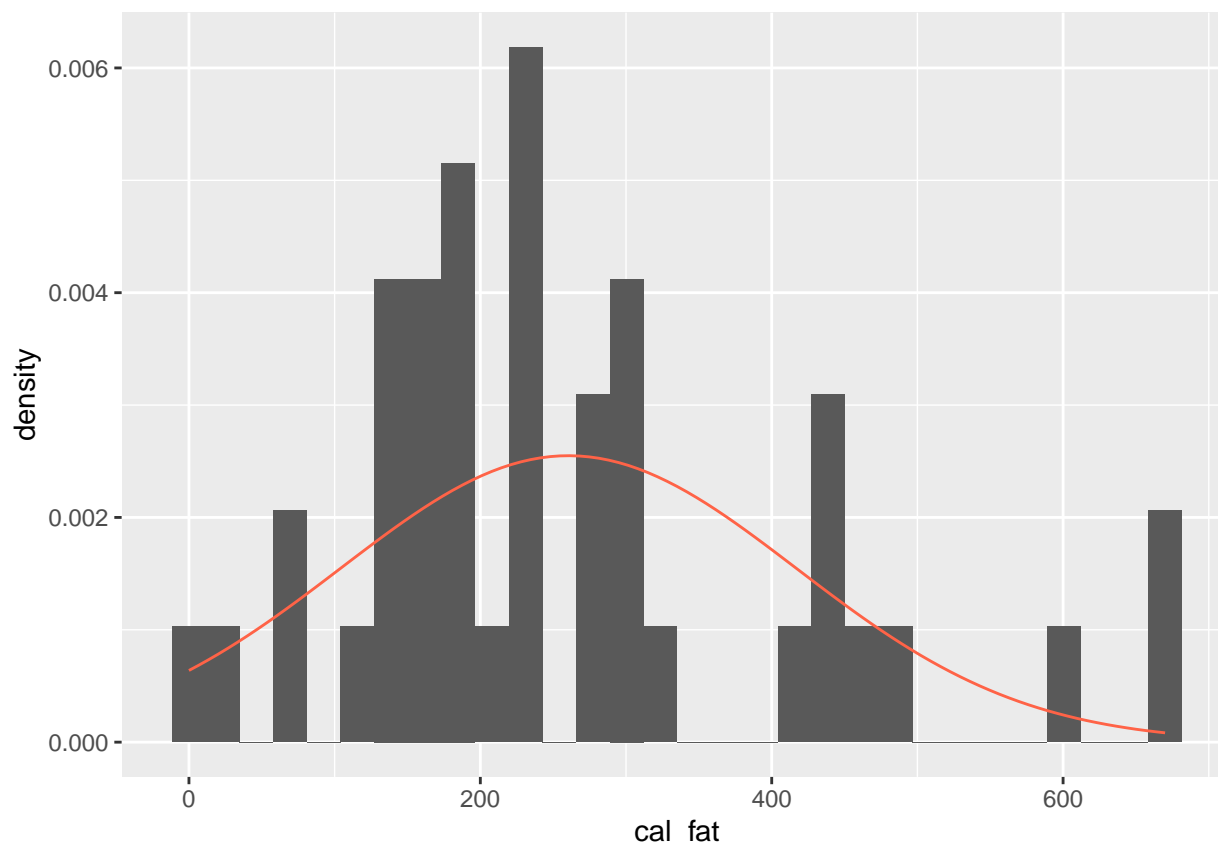
After initializing a blank plot with `geom_blank()`, the `ggplot2` package (within the `tidyverse`) allows us to add additional layers. The first layer is a density histogram. The second layer is a statistical function – the density of the normal curve, `dnorm`. We specify that we want the curve to have 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?

Based on the plot it does seem that the data follows a nearly normal distribution. The maximum density of the histogram is slightly higher than the density of the normal curve, but in terms of spread they are nearly the same.

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")
```
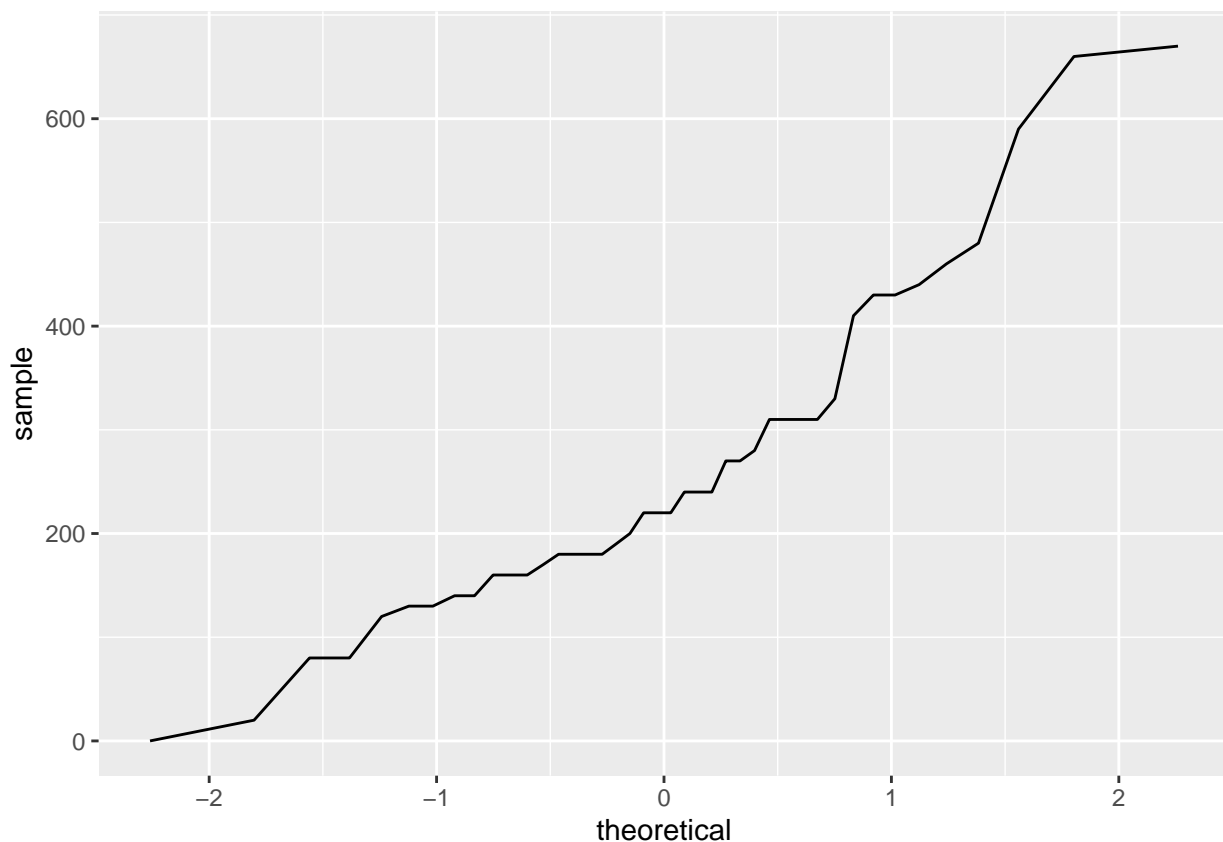

This time, you can use the `geom_line()` layer, while specifying that you will be creating a Q-Q plot with the `stat` argument. It's important to note that here, instead of using `x` instead `aes()`, you need to use `sample`.

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.)

```
df <- data.frame(sim_norm = sim_norm)
ggplot(data = df, aes(sample = sim_norm)) +
  geom_line(stat = "qq")
```
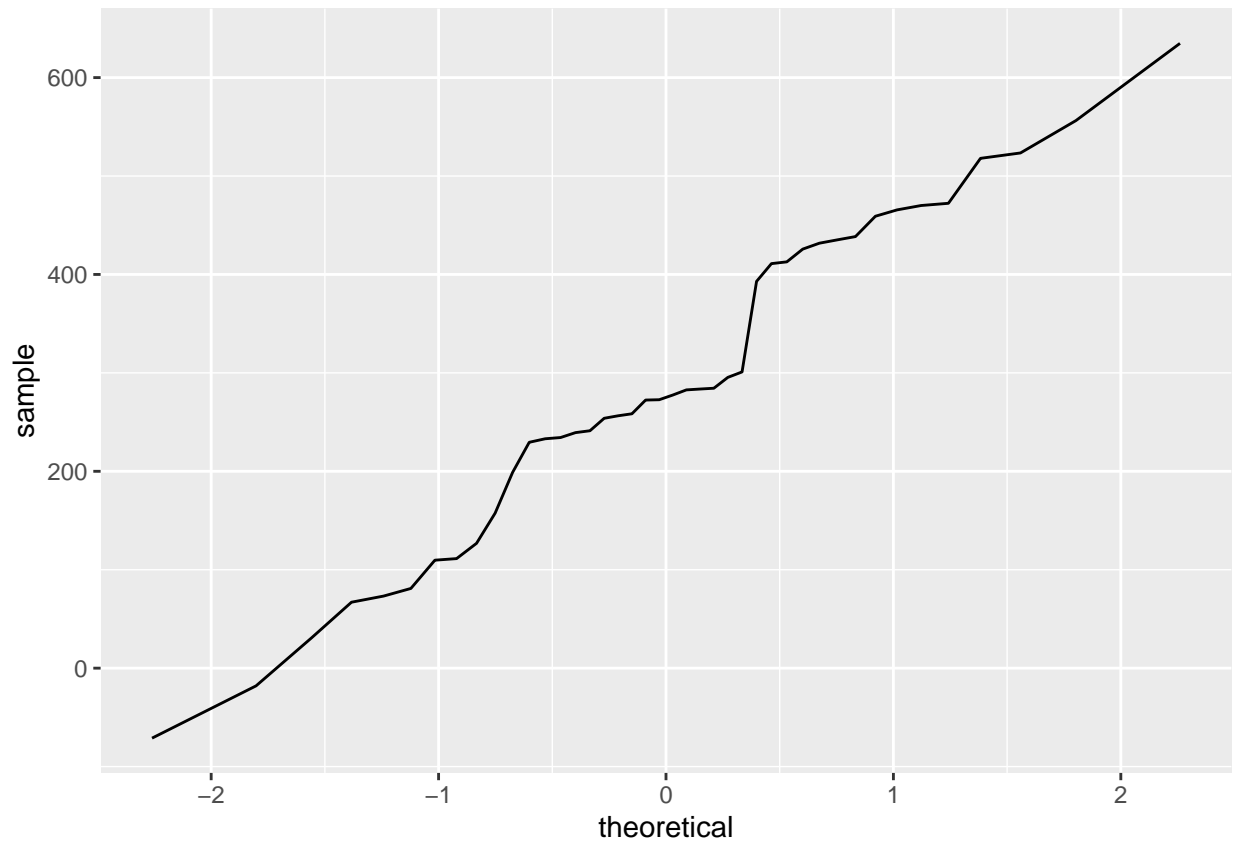
I got an error unless I did use the data argument, so I put `sim_norm` into a dataframe. The simulated normal distribution of dairy queen calories from fat follows a mostly diagonal line with some exceptions - some points are above or below the diagonal line. This plot is not much different then the probability plot for the real data. Based on this I reaffirm my statement that the dairy queen calories from fat follows mostly a normal distribution.

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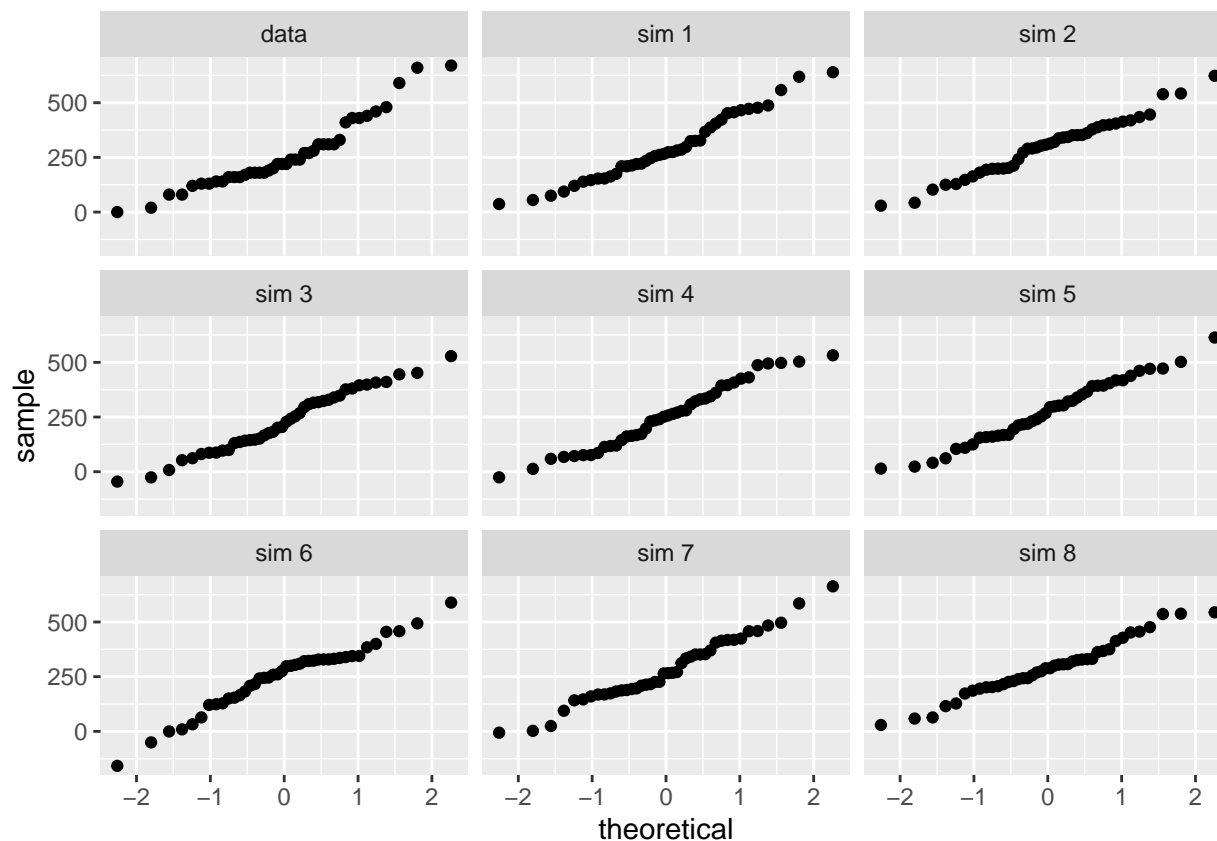
4. Does the normal probability plot for the calories from fat look similar to the plots created for the simulated data? That is, do the plots provide evidence that the calories are nearly normal?

This plot does provide evidence that the calories from fat from dairy queen are nearly normal. The slopes appear similar, they all follow basically a diagonal line with a few exceptions. The start and end of the line are the areas with more anomolies. The reason the start and the end of the line have more anomolies is that is where less points lie and they are a few sigmas out.

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

```
qqnormsim(sample = cal_fat, data = mcdonalds)
```
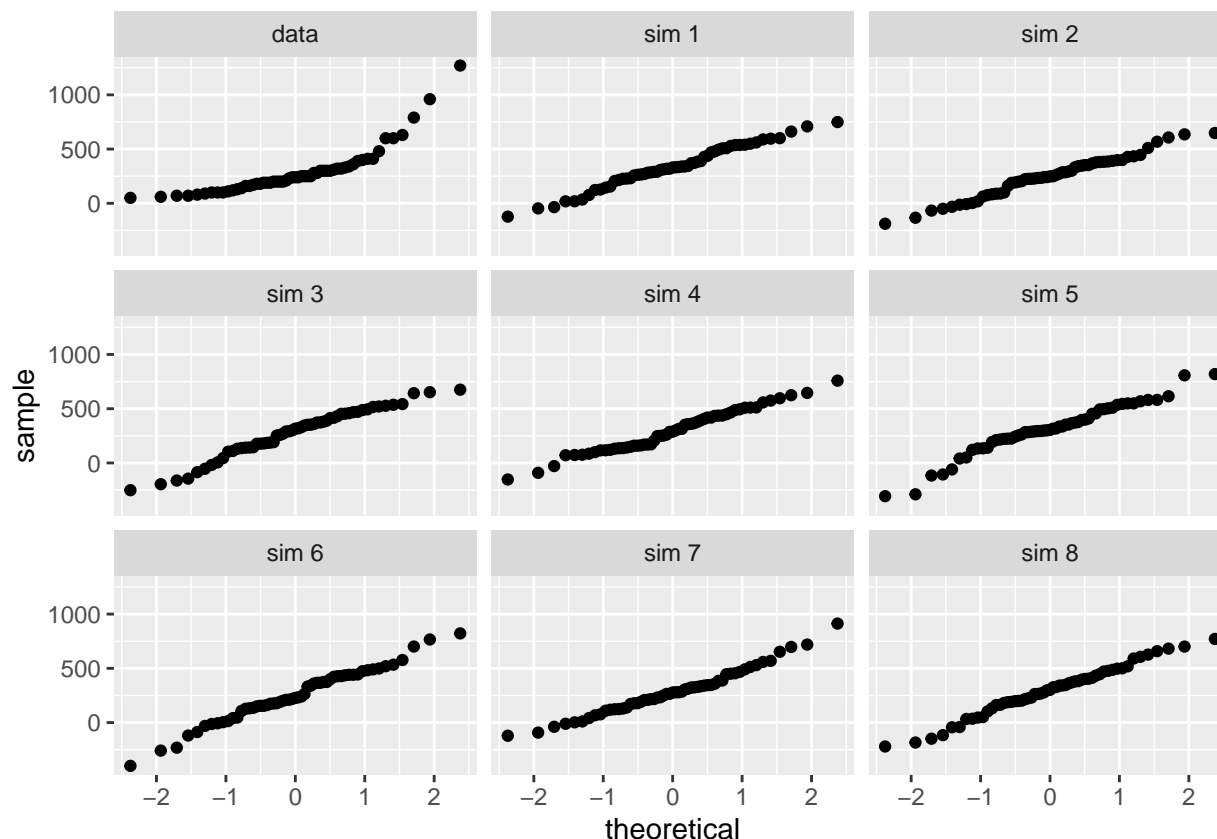
Based on these plot there is a predominant diagonal line but it also seems there is a slight curve - that is indicative of the right skew. I would still say it follows a mostly normal distribution with some skew and some anomalies at the ends. Anomolies at the ends happen on all of the plots so this does not affect my answer that it is a normal distribution with a right skew.

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.

9

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 x 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?

Probability question 1:

What is the probability that a randomly chosen subway product has less than 700 calories?

```
# Theoretical
subway <- fastfood %>%
  filter(restaurant == "Subway")
smean <- mean(subway$calories)
ssd <- sd(subway$calories)
t <- pnorm(q = 700, mean = smean, sd = ssd)
t
```

```
## [1] 0.7573991
```

```
# Empirical
e <- subway %>%
  filter(calories < 700) %>%
  summarise(percent = n() / nrow(subway))
e
```

```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1   0.719
```

```
t - e
```

```
##      percent
## 1 0.03864914
```

There is good agreement between the theoretical and empirical probabilties, 0.757 and 0.719 - only 0.0386 different.

Probability question 2:

What is the probability that a randomly chosen subway product has more than 25 grams of total fat?

```
# Theoretical
smean <- mean(subway$total_fat)
ssd <- sd(subway$total_fat)
t <- 1 - pnorm(q = 25, mean = smean, sd = ssd)
t
```

```
## [1] 0.3276727
```

```
# Empirical
e <- subway %>%
  filter(total_fat > 25) %>%
  summarise(percent = n() / nrow(subway))
e
```

```
## # A tibble: 1 x 1
##    percent
##      <dbl>
## 1    0.292
```

```
t - e
```

```
##       percent
## 1 0.03600605
```

There is good agreement between the theoretical and empirical probabilties, 0.327 and 0.292 - only 0.036 different.

The second probability question had greater agreement between the two methods as it had less of a difference between the theoretical and experimental probabilties.

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?

```
u <- unique(fastfood$restaurant)
for (i in 1:length(u)){
  restaurant <- fastfood %>%
    filter(restaurant == u[i])
  print(qqnormsim(sample = sodium, data = restaurant) +
    ggtitle(u[i]))
}
```
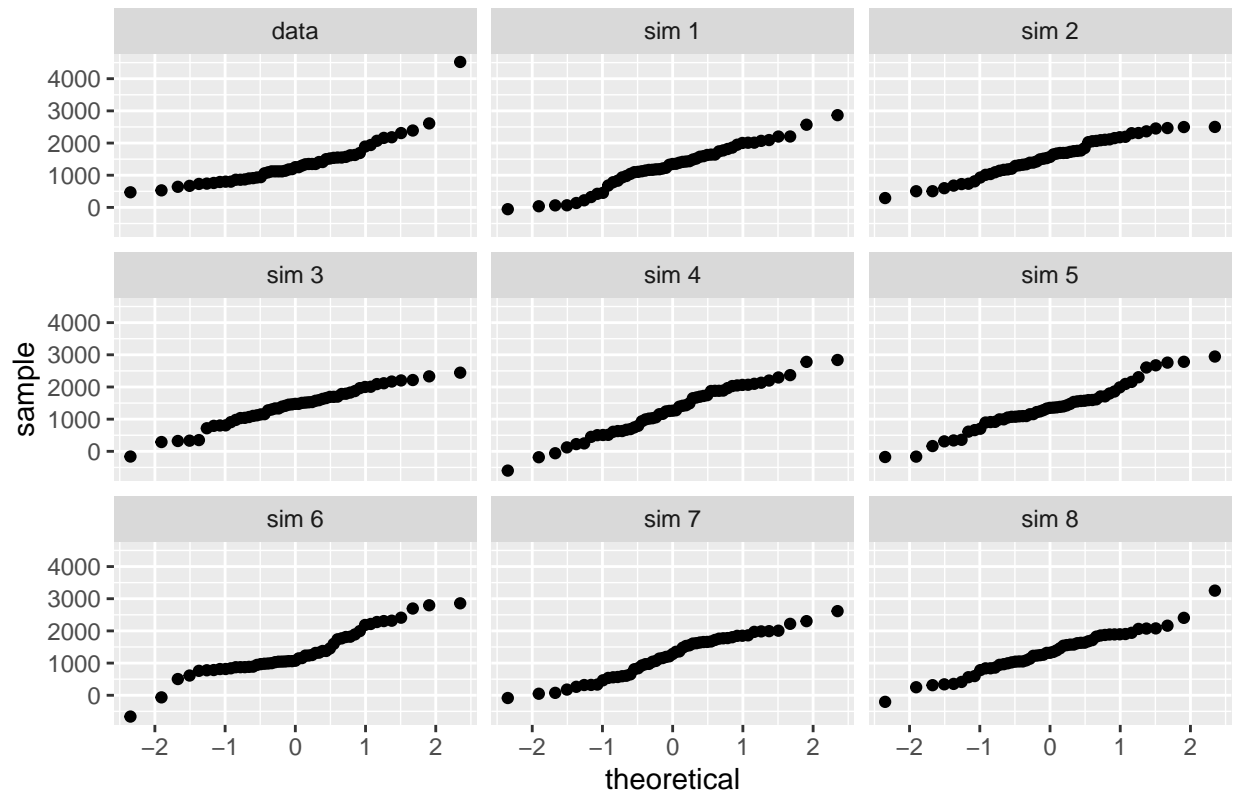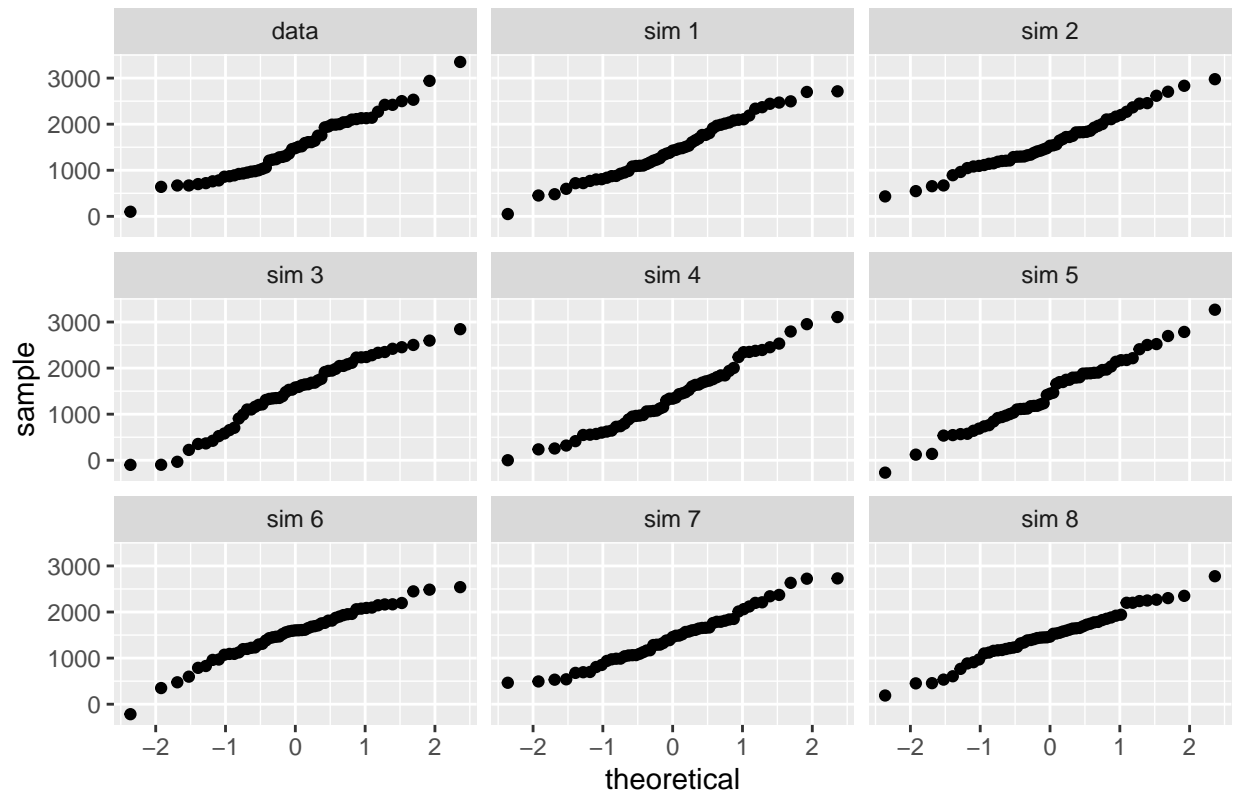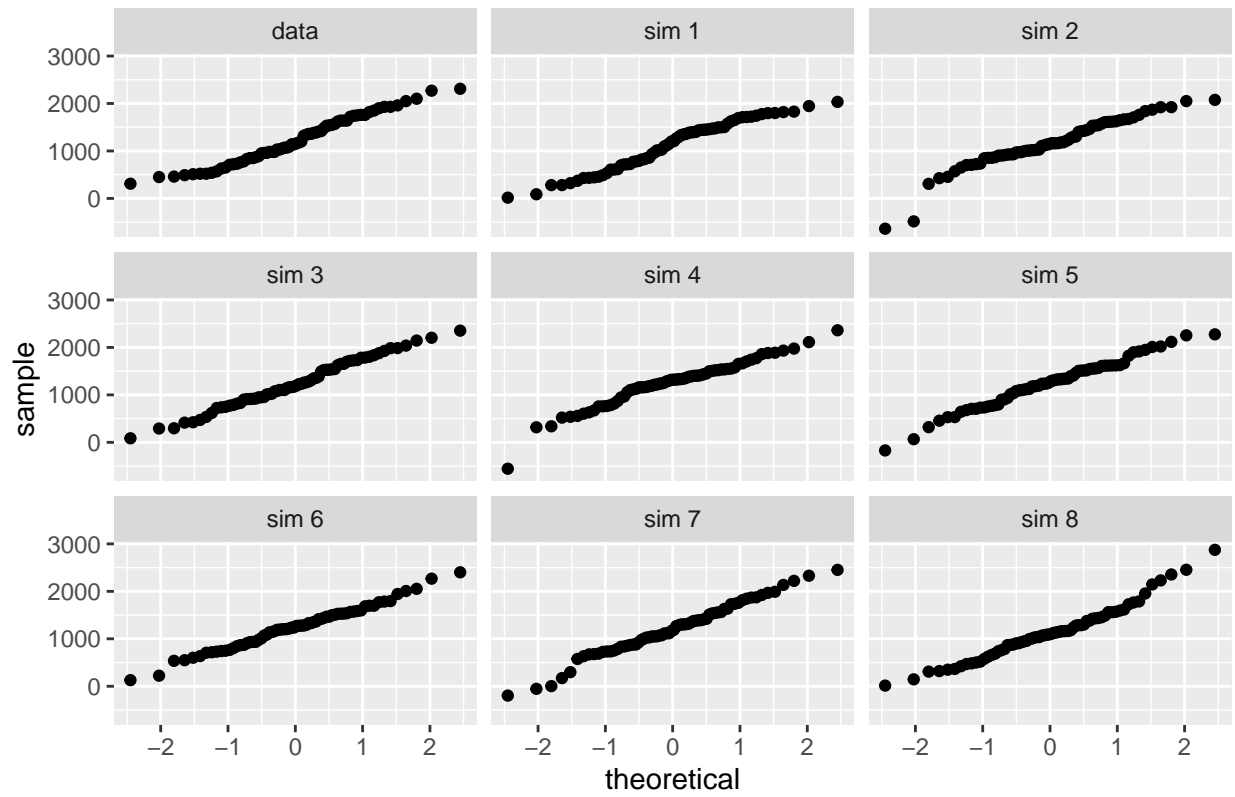
Mcdonalds
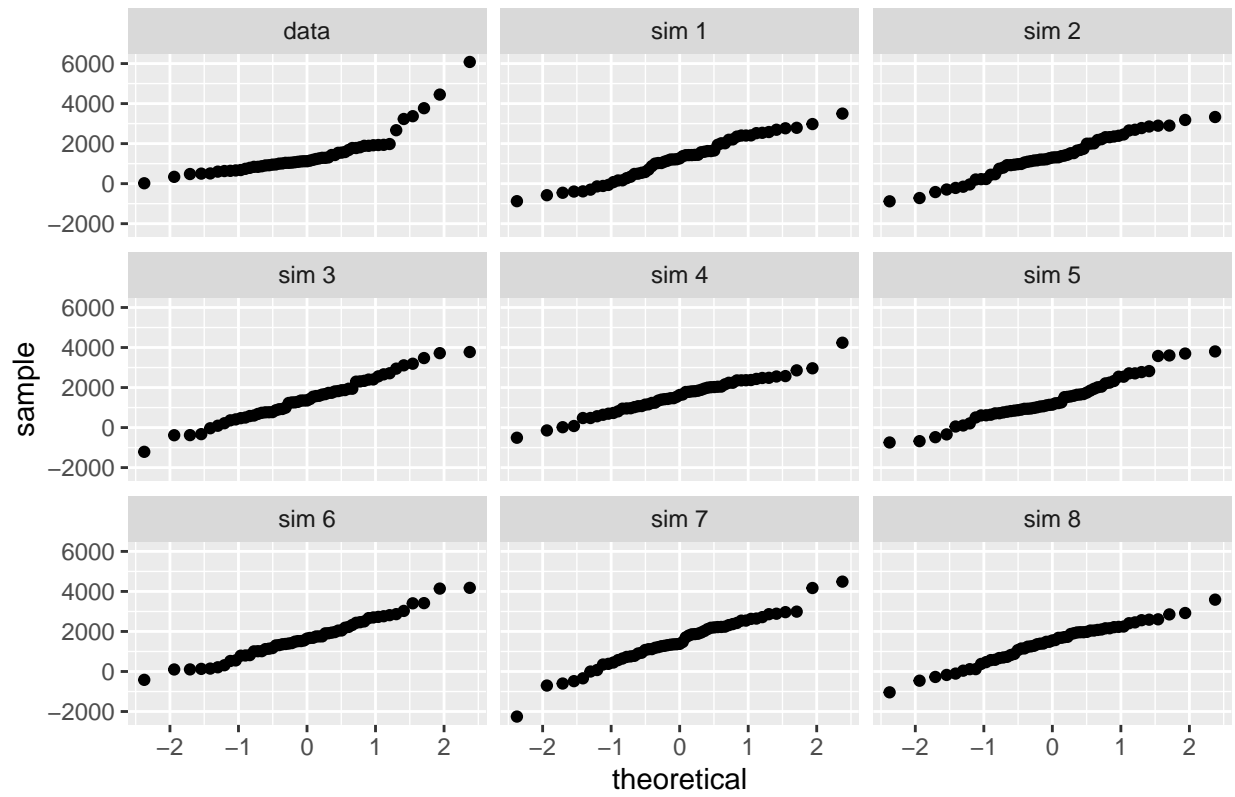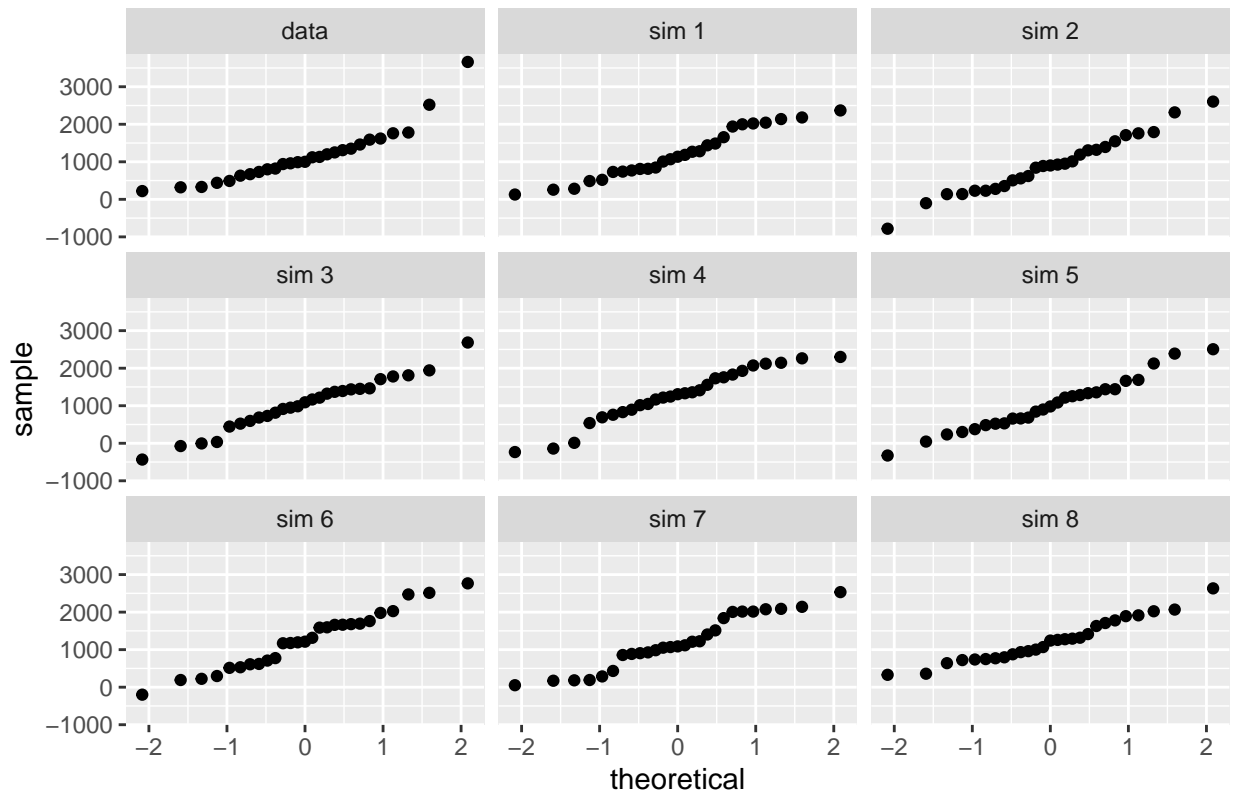
Sonic

Arbys
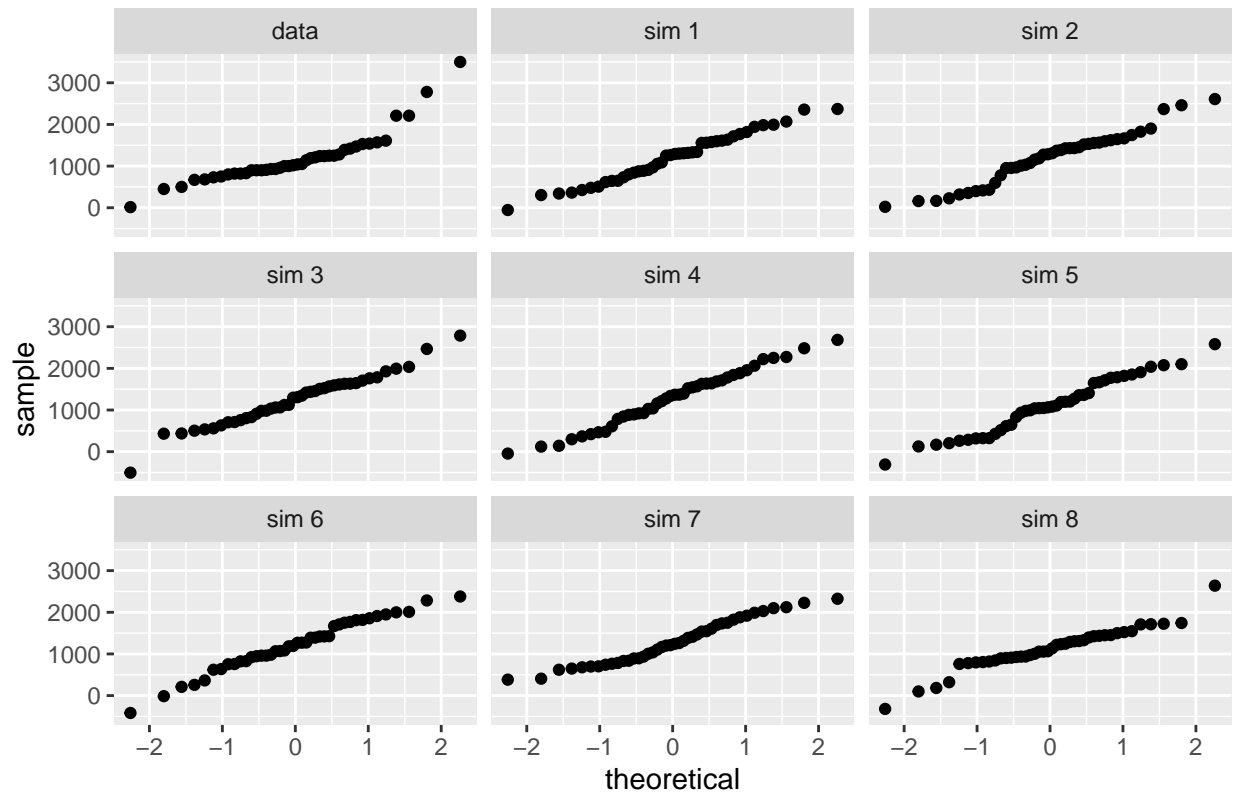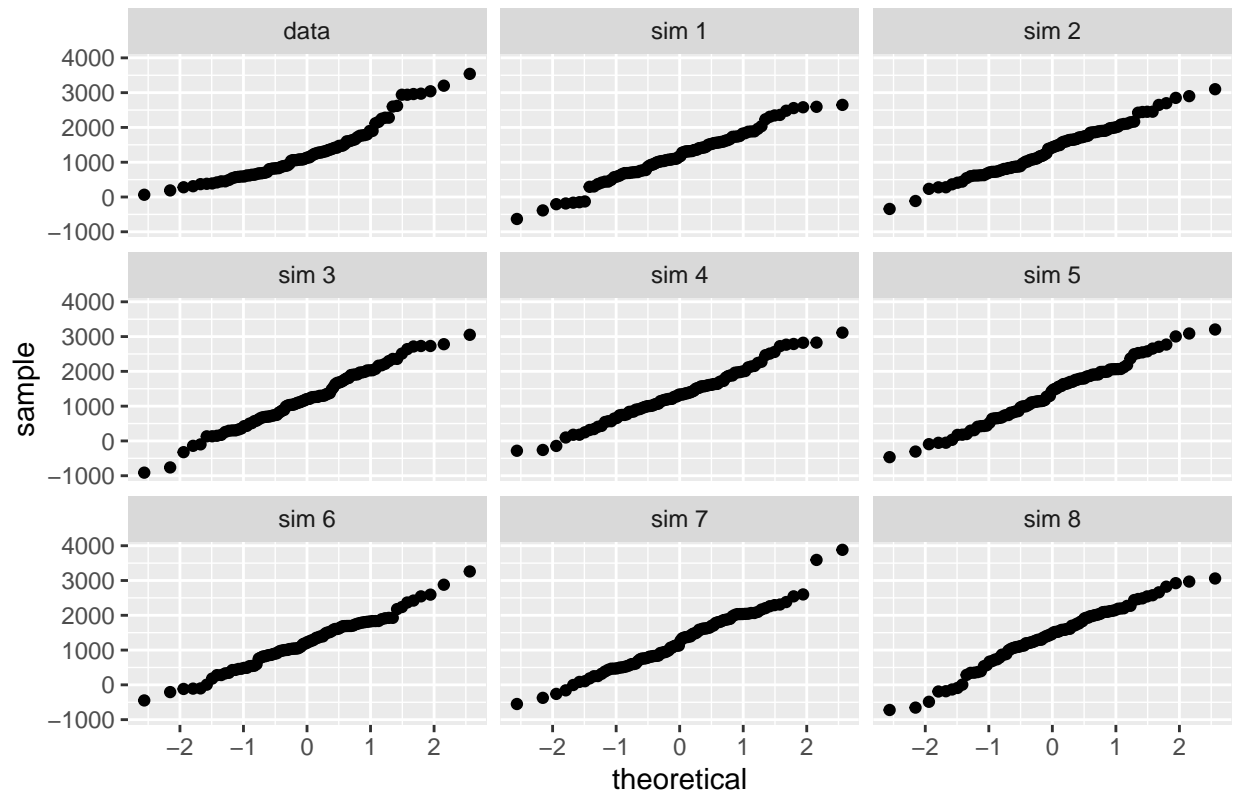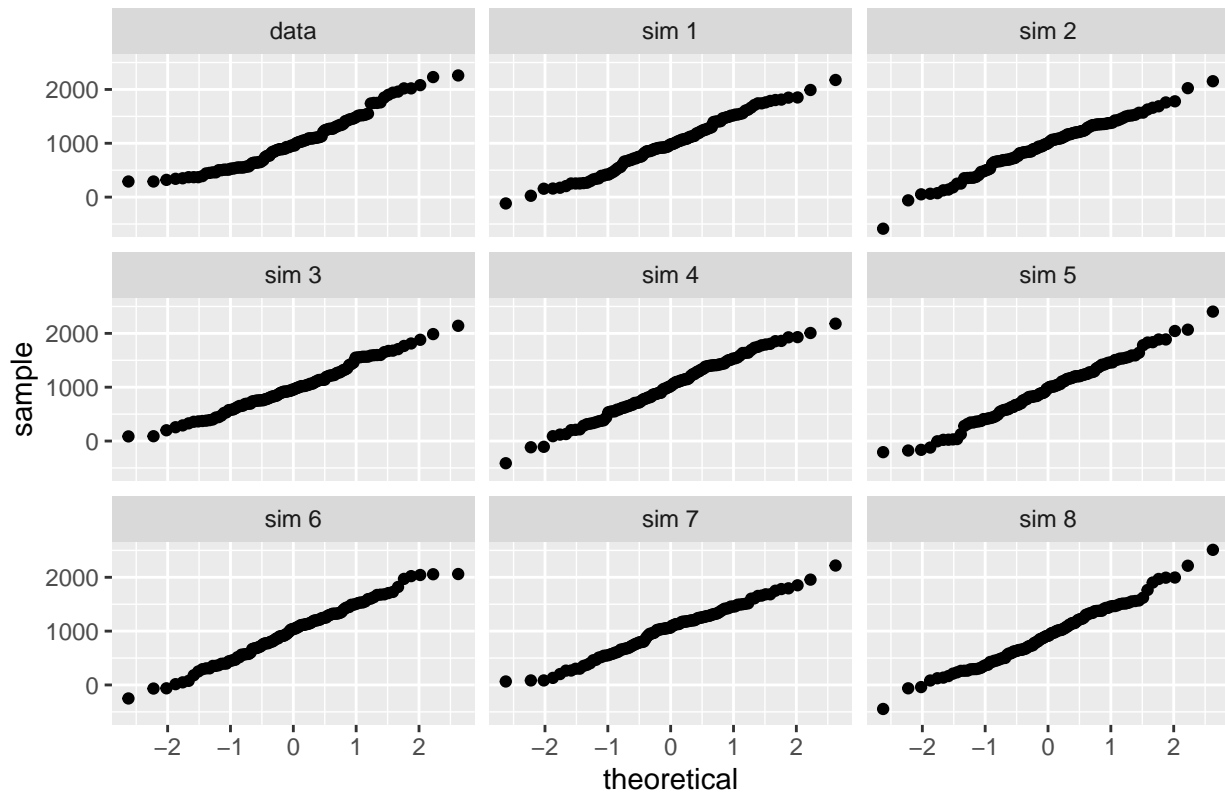
Burger King

Dairy Queen

Subway

Taco Bell

Based on these plots, most of them show a normal distribution for sodium - however the one that looks closest to normal is Burger King.

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?

Some of the plots for sodium may have a stepwise pattern if sodium is measured a certain way - say it is just 5mg increments 5mg, 10mg, 15mg. . . measuring sodium this way rather than continuously would yield a stepwise pattern.

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.

Make the normal probability plot.

```
restaurant <- fastfood %>%
    filter(restaurant == "Mcdonalds")
qqnormsim(sample = total_carb, data = restaurant)
```
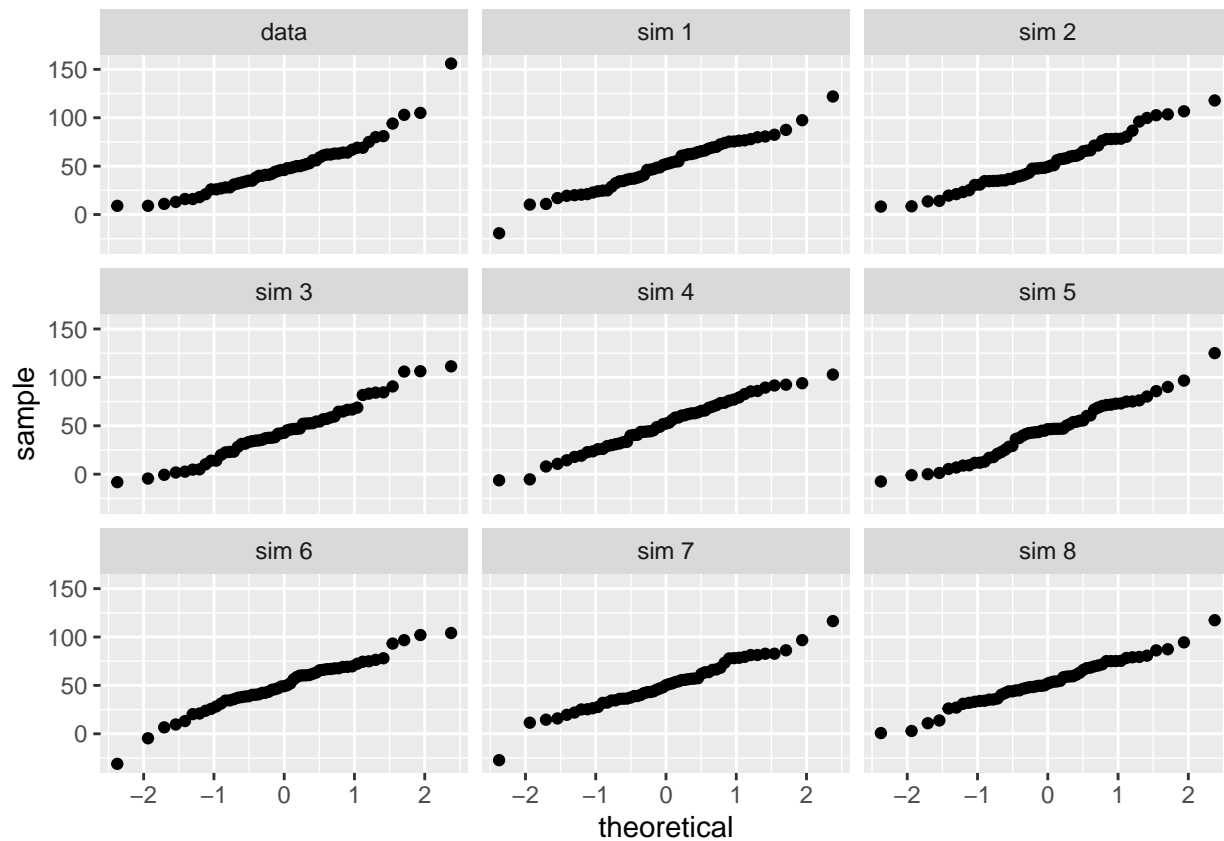
Based on this plot, the data appears right skewed.

Make the histogram.

```
smean <- mean(restaurant$total_carb)
ssd <- sd(restaurant$total_carb)
ggplot(data = restaurant, aes(x = total_carb)) +
        geom_blank() +
        geom_histogram(aes(y = ..density..)) +
        stat_function(fun = dnorm, args = c(mean = smean, sd = ssd), col = "tomato")
```
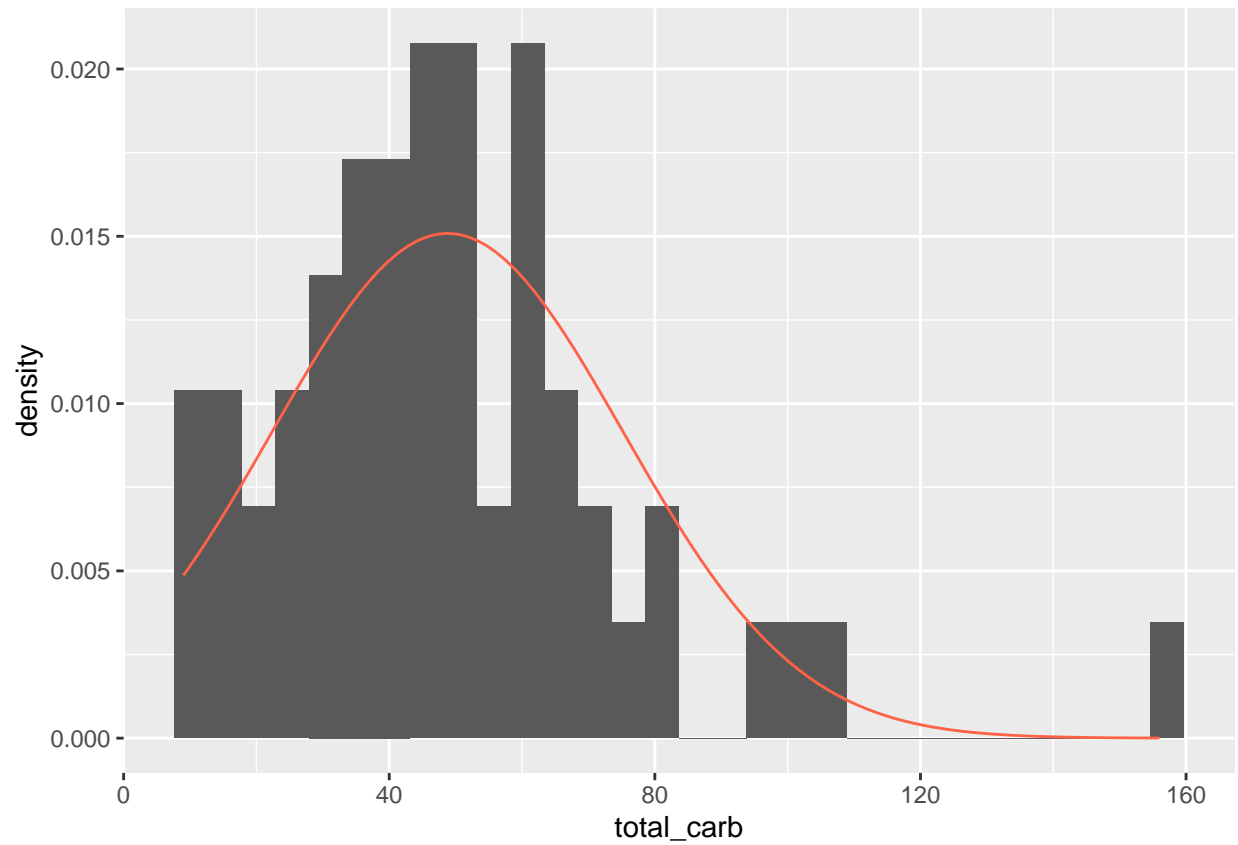
The total carb data from Mcodnalds follows a normal distribution but has a skew to the right. * * *