Distributions of Random Variables

Zachary Palmore

2020-09-27

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

Pre-exercise

Checking for the data:

```
head(fastfood)
```

```
## # A tibble: 6 x 17
##   restaurant item  calories cal_fat total_fat sat_fat trans_fat cholesterol
##   <chr>      <chr>    <dbl>   <dbl>     <dbl>   <dbl>     <dbl>       <dbl>
## 1 Mcdonalds  Arti~      380      60         7       2       0            95
## 2 Mcdonalds  Sing~      840     410        45      17       1.5         130
## 3 Mcdonalds  Doub~     1130     600        67      27       3           220
## 4 Mcdonalds  Gril~      750     280        31      10       0.5         155
## 5 Mcdonalds  Cris~      920     410        45      12       0.5         120
## 6 Mcdonalds  Big ~      540     250        28      10       1            80
## # ... with 9 more variables: sodium <dbl>, total_carb <dbl>, fiber <dbl>,
## #   sugar <dbl>, protein <dbl>, vit_a <dbl>, vit_c <dbl>, calcium <dbl>,
## #   salad <chr>
```

Minor change

Instead of this,

```
# mcdonalds <- fastfood %>%
#  filter(restaurant == "McDonalds")
```

I used,

```
# A lower case d was changed from "McDonalds"
# based on observations in the data
mcdonalds <- fastfood %>%
  filter(restaurant == "Mcdonalds")
dairy_queen <- fastfood %>%
  filter(restaurant == "Dairy Queen")
```

to filter observations for the restaurant McDonalds. The previous version gave me zero observations.

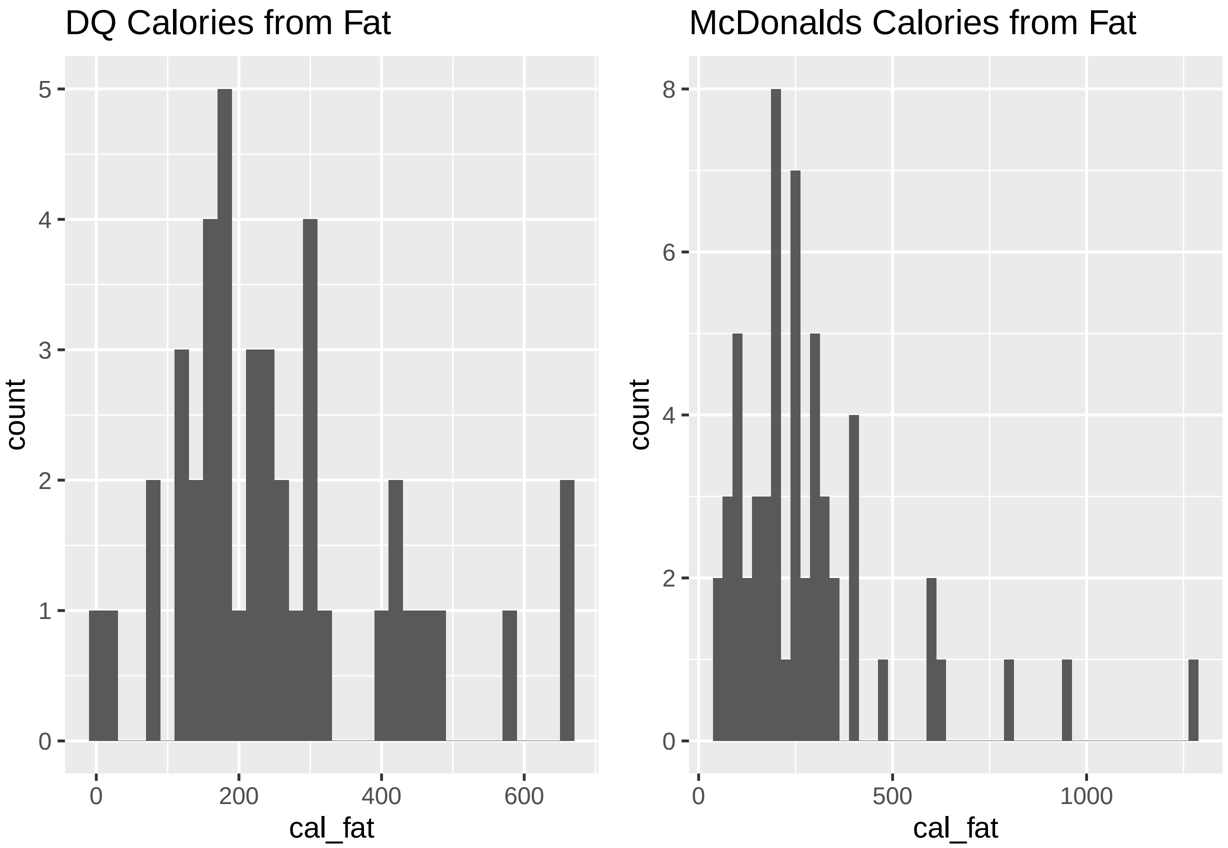
```
view(mcdonalds)
```

Exercise 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?

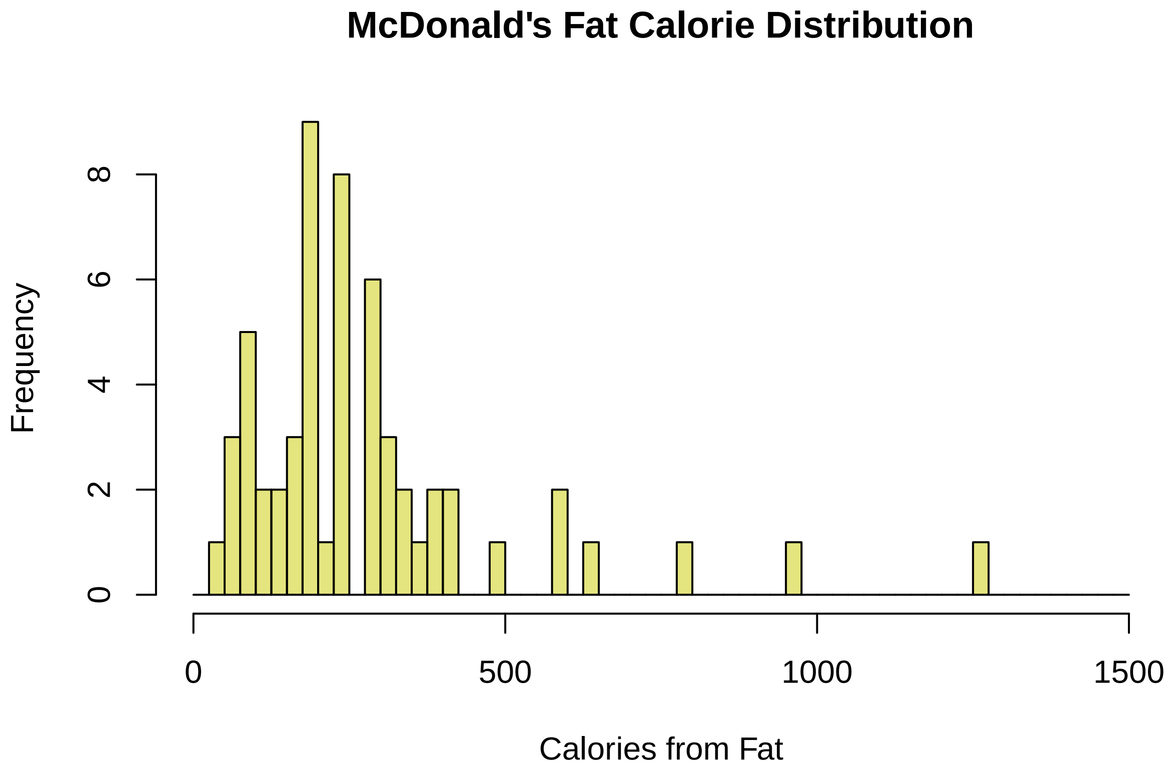
Let's see, using ggplot we can see a difference in the spreads of the distributions. It appears McDonald's spread is much larger and that there is a higher frequency of fats from calories under 500, but another visual might be better.

```
histdq_calfat <- ggplot(data = dairy_queen, aes(x = cal_fat)) +
        geom_histogram(binwidth = 20) + ggtitle("DQ Calories from Fat")
histmcd_calfat <-  ggplot(data = mcdonalds, aes(x = cal_fat)) +
        geom_histogram(binwidth = 25) + ggtitle("McDonalds Calories from Fat")
ggarrange(histdq_calfat, histmcd_calfat)
```

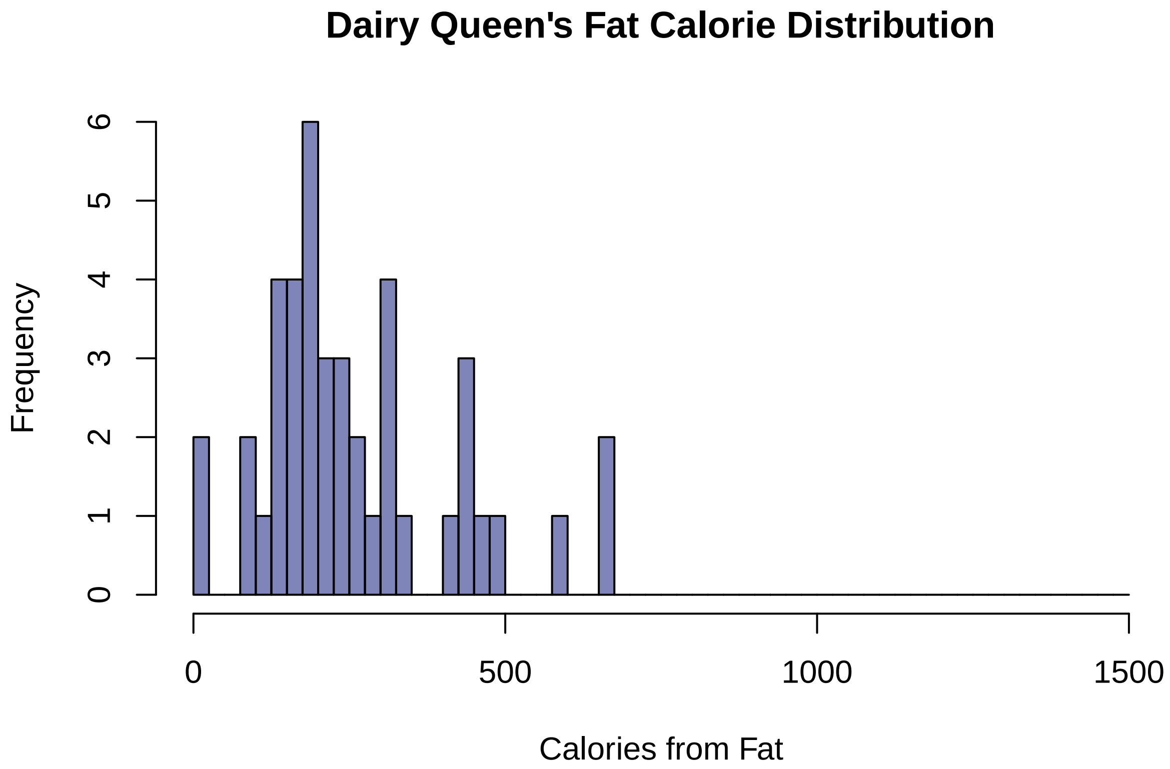


Without ggplot, let's see if we can better visualize both distributions. Adding some color would also be helpful.

```
hist(mcdonalds$cal_fat,
     col = rgb(red = 0.80, green = 0.80, blue = 0, alpha = 0.50),
     xlab = "Calories from Fat",
     main = "McDonald's Fat Calorie Distribution",
     breaks = seq(0,1500,25))
```

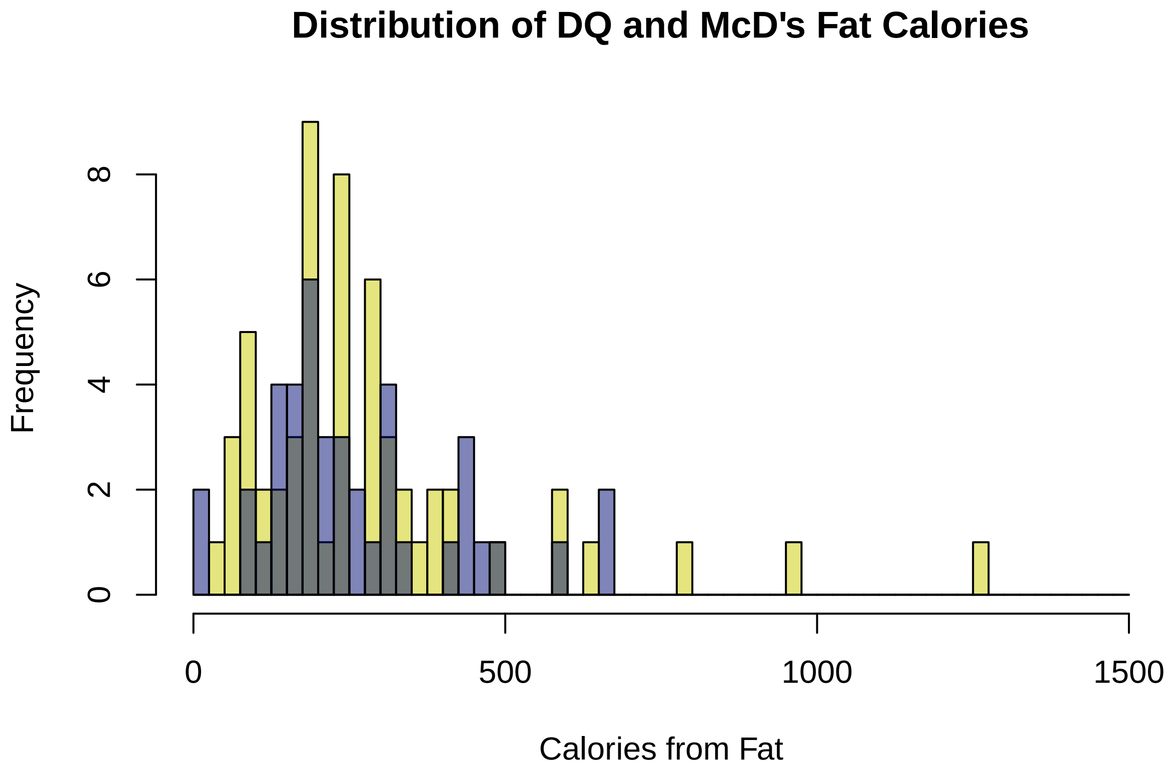


```r
hist(dairy_queen$cal_fat,
     col = rgb(red = 0, green = 0.05, blue = 0.45, alpha = 0.50),
     xlab = "Calories from Fat",
     main = "Dairy Queen’s Fat Calorie Distribution",
     breaks = seq(0,1500,25))
```



With both x-axes set to the same range (0 to 1500) we can see very clearly the spread of McDonald's calories from fat extends beyond 1000, while Dairy Queen's stays under 750. Both histograms also have their centers concentrated just under 250 calories with their distribution's shapes skewed to the right.

```
# or potentially a better way to visualize
mcd_hist_calfat <- hist(mcdonalds$cal_fat, plot = FALSE, breaks = seq(0,1500,25))
dq_hist_calfat <- hist(dairy_queen$cal_fat, plot = FALSE, breaks = seq(0,1500,25))
plot(mcd_hist_calfat,
     col = rgb(red = 0.80, green = 0.80, blue = 0, alpha = 0.50),
     xlab = "Calories from Fat",
     main = "Distribution of DQ and McD's Fat Calories")

plot(dq_hist_calfat, add = TRUE,
     col = rgb(red = 0, green = 0.05, blue = 0.45, alpha = 0.50))
```



Here, where both restaurants are on the same histogram, we can also see that McDonald's calories from fat are more frequent between 0 and 500 when compared to Dairy Queen. It is also easier to see McDonald's distribution skewed to the right much more than Dairy Queen.

```
# McDonalds fat calorie Distribution
summary(mcdonalds$cal_fat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    50.0   160.0   240.0   285.6   320.0  1270.0
```

```
# Dairy Queen fat calorie Distribution
summary(dairy_queen$cal_fat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   160.0   220.0   260.5   310.0   670.0
```

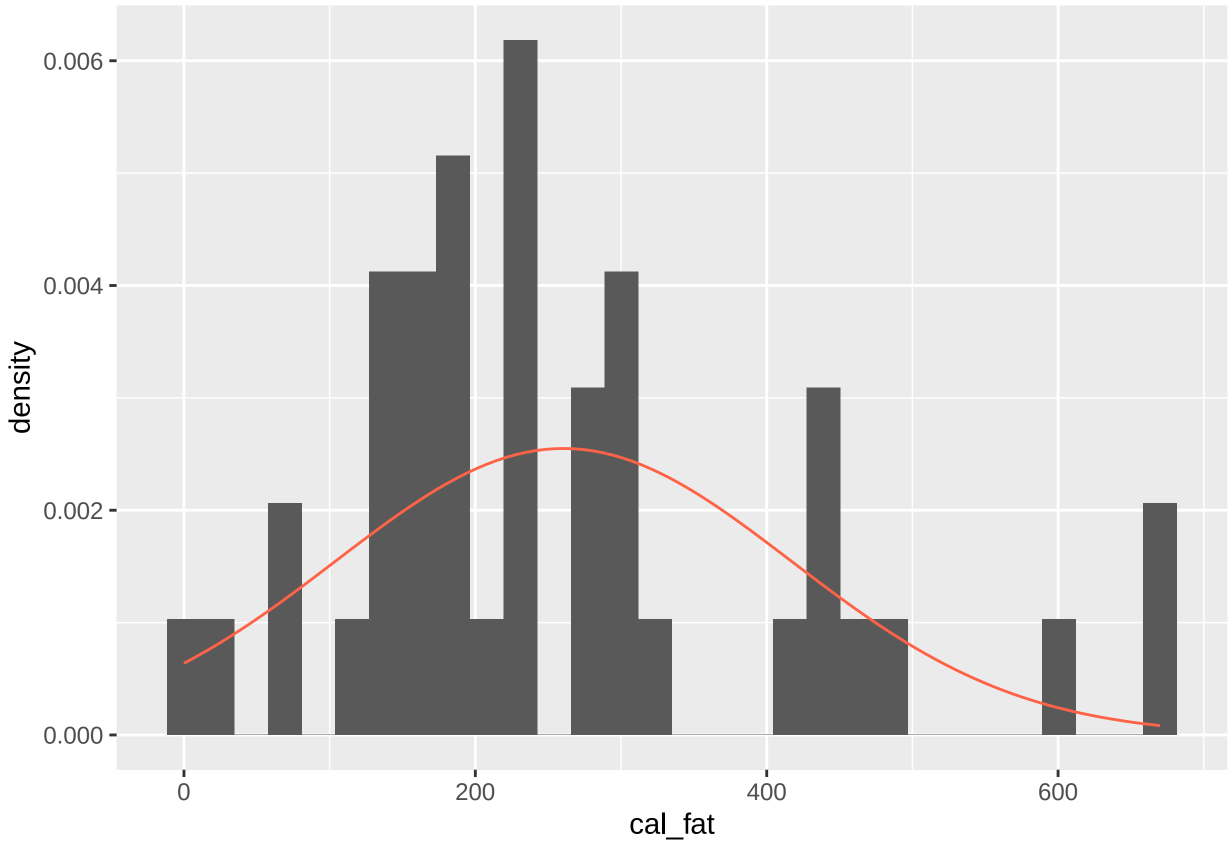
Exercise 2

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

Loading plot…

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

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



Yes, based on this plot it appear as though the fat calories from Dairy Queen follow a nearly normal distribution.

Exercise 3

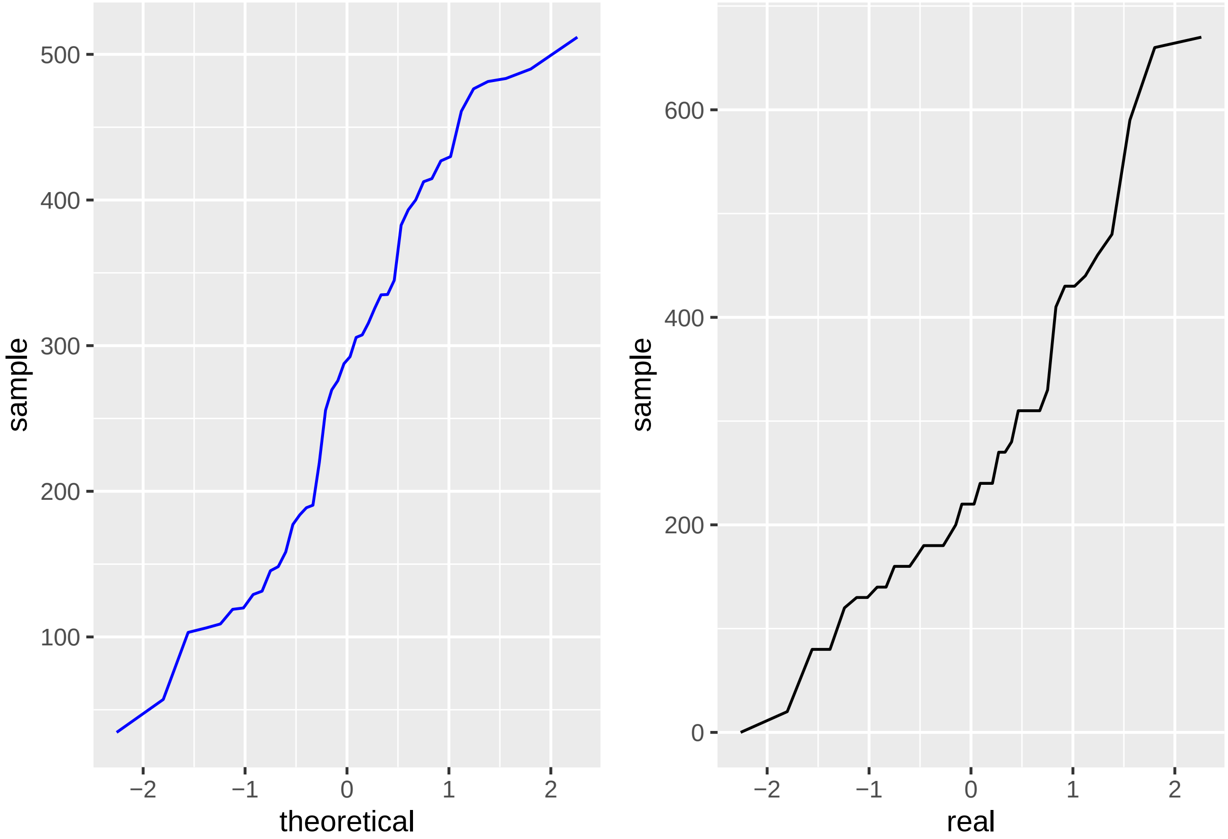
Make a normal probability plot of sim_norm. Do all of the points fall on the line? How does this plot compare to the probability plot for the real data? (Since sim_norm is not a dataframe, it can be put directly into the sample argument and the data argument can be dropped.)

From lab we have:

```
set.seed(09182020)
# QQ plot creation
qqplot_dq <- ggplot(data = dairy_queen, aes(sample = cal_fat)) +
  geom_line(stat = "qq")
# Simulated normal distribution us DQ stats
sim_norm <- rnorm(n = nrow(dairy_queen), mean = dqmean, sd = dqsd)
```

Using the simulated normal distribution to create a true normal probability plot for comparison.

```
simqqplot_dq <- ggplot(data = dairy_queen, aes(sample = sim_norm)) +
  geom_line(stat = "qq", col = "blue")
qqplot_dq <- ggplot(data = dairy_queen, aes(sample = cal_fat)) +
  geom_line(stat = "qq") + labs(x = "real")
ggarrange(simqqplot_dq, qqplot_dq)
```



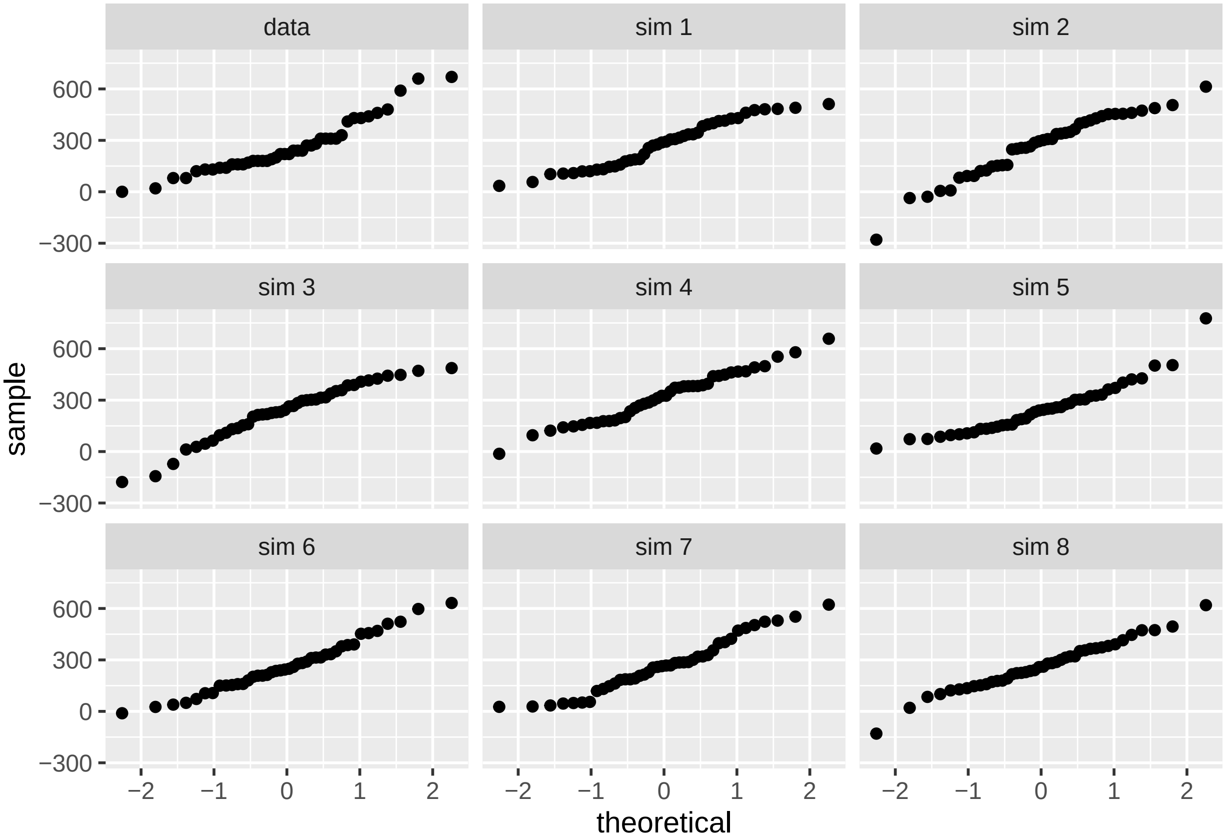
There is a drop in the center of the real distribution of dairy queen's calories from fat and it does not show this drop on the theoretical distribution. The theoretical sample also does not continue as far (up and to the right) on the y-axis after 2, whereas the real data does.

Exercise 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 female heights are nearly normal?

Reviewing the simulations:

```
# Lab code
set.seed(9182020)
qqnormsim(sample = cal_fat, data = dairy_queen)
```



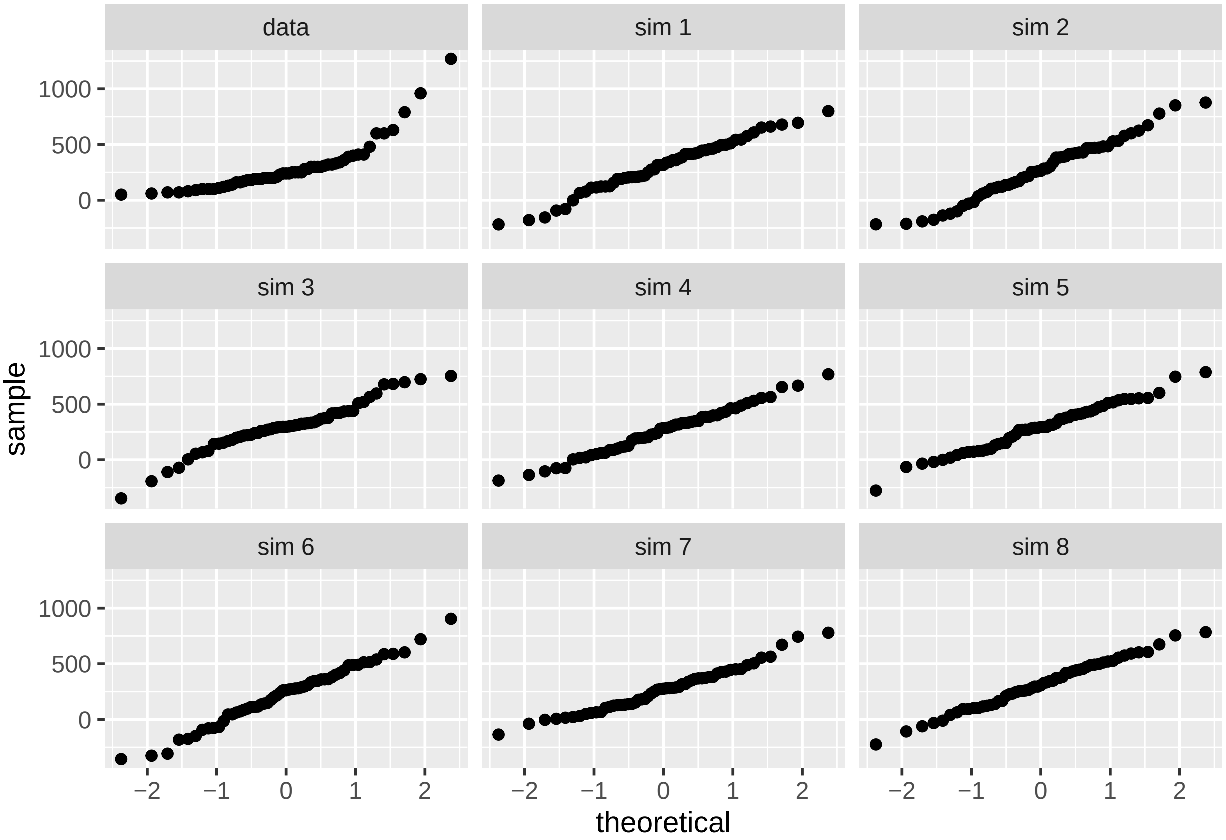
(Substituting "the data" for female heights) To some extent, yes, the plots provide evidence that the data is nearly normal. However, a closer review might be necessary as there are jumps (or breaks) in the real data that do not appear normal. When compared to the simulations, only a few have clear jumps in the data. This provides evidence against a normal distribution.

Exercise 5

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

Repeating this process for McDonalds calories.

```
set.seed(09192020)
qqnormsim(sample = cal_fat, data = mcdonalds)
```



In this case, the data does not appear normally distributed. The theoretical simulations are all lined up on or near a straight diagonal line, whereas, the McDonalds data is not. It takes on more of a "J" shape and has a few points that extend above the threshold where most simulations stop plotting new points. In general, this threshold is at about 750 - 800 on the y-axis.

Additionally, the bulk of McDonalds data is concentrated in the range of 0 - 500. In the simulations the data nearly always continues just beyond 500 in straight diagonal path equivalent to a normal distribution. McDonalds strays from this path, dipping lower than normal and seemingly maintaining that slope, until within a few points it over-corrects and misses the normal distribution again.

Exercise 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?

Viewing lab functions:

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

```
## [1] 0.01501523
```

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

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

```
# abs(0.01501523 - 0.04761905)  = 0.03260382
```

In this given example, the absolute difference between the theoretical normal and empirical methods was 0.033.

Since McDonalds does not appear to have as normal of a distribution, the difference between their theoretical and empirical calculations should be larger than Dairy Queen. To find out, I will run the same calculations to compare the theoretical normal and empirical probabilities of selecting (at random) a menu item with calories from fat at less than 600 calories.

```
mcdmean <- mean(mcdonalds$cal_fat)
mcdsd <- sd(mcdonalds$cal_fat)
1 - pnorm(q = 600, mean = mcdmean, sd = mcdsd)
```

```
## [1] 0.07733771
```

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

```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1  0.0702
```

```
# abs(0.07733771 - 0.07017544) = 0.00716227
```

The difference between the theoretical normal and empirical methods was 0.007. This is interesting because theoretical and empirical probabilities are closer to agreement in this scenario than in Dairy Queen's. Specifically, 0.007 is a smaller difference than the given example of 0.033. Yet, the distributions of the data overall are different, with McDonalds being farther from normal than Dairy Queen. Is this the same for other McDonald's scenarios?

```
# at greater than 200 calories of fat
1 - pnorm(q = 200, mean = mcdmean, sd = mcdsd)
```

```
## [1] 0.650833
```

```
mcdonalds %>%
  filter(cal_fat > 200) %>%
  summarise(percent = n() / nrow(mcdonalds))
```

```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1   0.561
```

```
# abs(0.650833 - 0.5614035) = 0.0894295
```

Looking at the probabilities of finding a menu item greater than 200 calories of fat, does produce a difference greater than our Dairy Queen example at 0.089 (McDonalds) to 0.033.

```
# at greater than 100 calories of fat
1 - pnorm(q = 100, mean = mcdmean, sd = mcdsd)
```

```
## [1] 0.7996202
```

```
mcdonalds %>%
  filter(cal_fat > 100) %>%
  summarise(percent = n() / nrow(mcdonalds))
```

```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1   0.842
```

```
# abs(0.8421053 - 0.7996202) = 0.0424851
```

Here again, the probabilities of finding a menu item greater than 100 calories of fat, produces a difference greater than our Dairy Queen example of 0.042 (McDonalds) to 0.033.

```
# at less than 800 calories of fat
pnorm(q = 800, mean = mcdmean, sd = mcdsd)
```

```
## [1] 0.9900599
```

```
mcdonalds %>%
  filter(cal_fat < 800) %>%
  summarise(percent = n() / nrow(mcdonalds))
```

```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1   0.965
```

```
# abs(0.9900599 - 0.9649123) = 0.0251476
```

However, when considering the probabilities of how many items contain less than 800 calories from fat we again see a difference in probabilities that is smaller than the given example.

```
# at less than 1000 calories of fat
pnorm(q = 900, mean = mcdmean, sd = mcdsd)
```

```
## [1] 0.9972929
```

```r
mcdonalds %>%
  filter(cal_fat < 900) %>%
  summarise(percent = n() / nrow(mcdonalds))
```

```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1   0.965
```

```r
# abs(0.9972929 - 0.9649123) =  0.0323806
```

The difference in theoretical and empirical probabilities of finding a menu item with calories from fat that are less than 900 at McDonalds is 0.032. This is very close to the given example (0.033), indicating that both methods were closer in agreement.

When thinking about the big picture of each distribution, each of these differences in agreement makes sense. Since McDonalds has tail of data skewing it towards menu items of higher amounts of calories from fat, the theoretical and empirical probabilities will be depend on whether or not these data are included in the distribution (ex: by using greater than or less than symbols to include and exclude data in the probability calculation).

I also wonder what the probability of selecting a menu item at less than 500 calories from fat at Sonic is if selected at random. Given that one large container of fries (at Mcdonalds) is estimated to be just under 500 calories, I thought it might be interesting to see for another restaurant.

```r
sonic <- fastfood %>%
  filter(restaurant == "Sonic")
sonicmean <- mean(sonic$cal_fat)
sonicsd <- sd(sonic$cal_fat)
1 - pnorm(q = 500, mean = sonicmean, sd = sonicsd)
```

```
## [1] 0.206046
```

```r
sonic %>%
  filter(cal_fat > 500) %>%
  summarise(percent = n() / nrow(sonic))
```

```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1   0.189
```

```r
# abs(0.793954 - 0.8113208) =  0.0173668 (with less than 500 calories from fat)
# or with greater than 500 calories from fat
# abs(0.206046 - 0.1886792) = 0.0173668
```

With an absolute difference between the theoretical normal and empirical methods of 0.017 it appears the theoretical and empirical probabilities of these are also closer to agreement than the example scenario.

As a reminder, the example Dairy Queen scenario found that the theoretical and empirical probabilities of a menu item containing greater than 600 calories of fat were 0.015 and 0.048 respectively. The difference between these two is approximately 0.033. The probabilities of finding a menu item at Sonic greater than 500 calories from fat were 0.206 theoretical) and 0.189 (empirical). The difference between these is approximately 0.017 which is a smaller difference (and thus closer agreement) than the example difference of 0.033.

Exercise 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?

There 8 restaurants, with the following names:

```
unique(fastfood$restaurant)
```

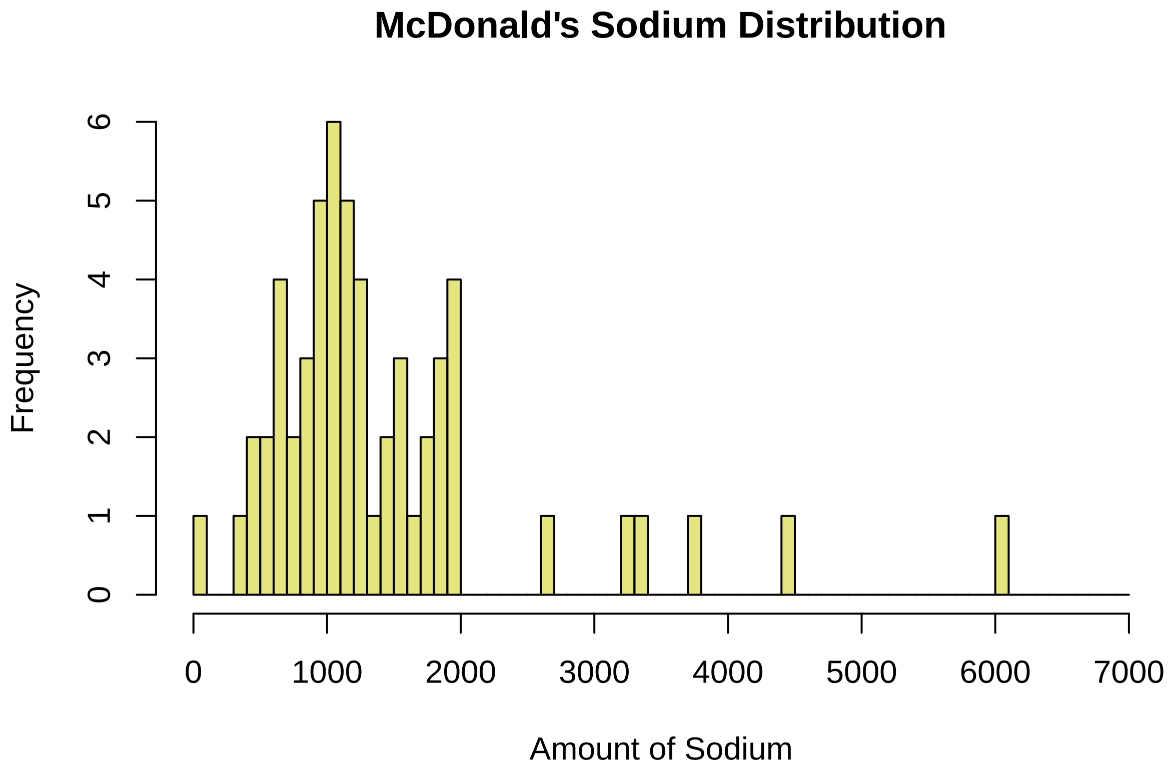
```
## [1] "Mcdonalds"   "Chick Fil-A" "Sonic"       "Arbys"       "Burger King"
## [6] "Dairy Queen" "Subway"      "Taco Bell"
```

It makes sense to select out each restaurant to isolate the data needed for creating their distributions and comparing each to a normal distribution.

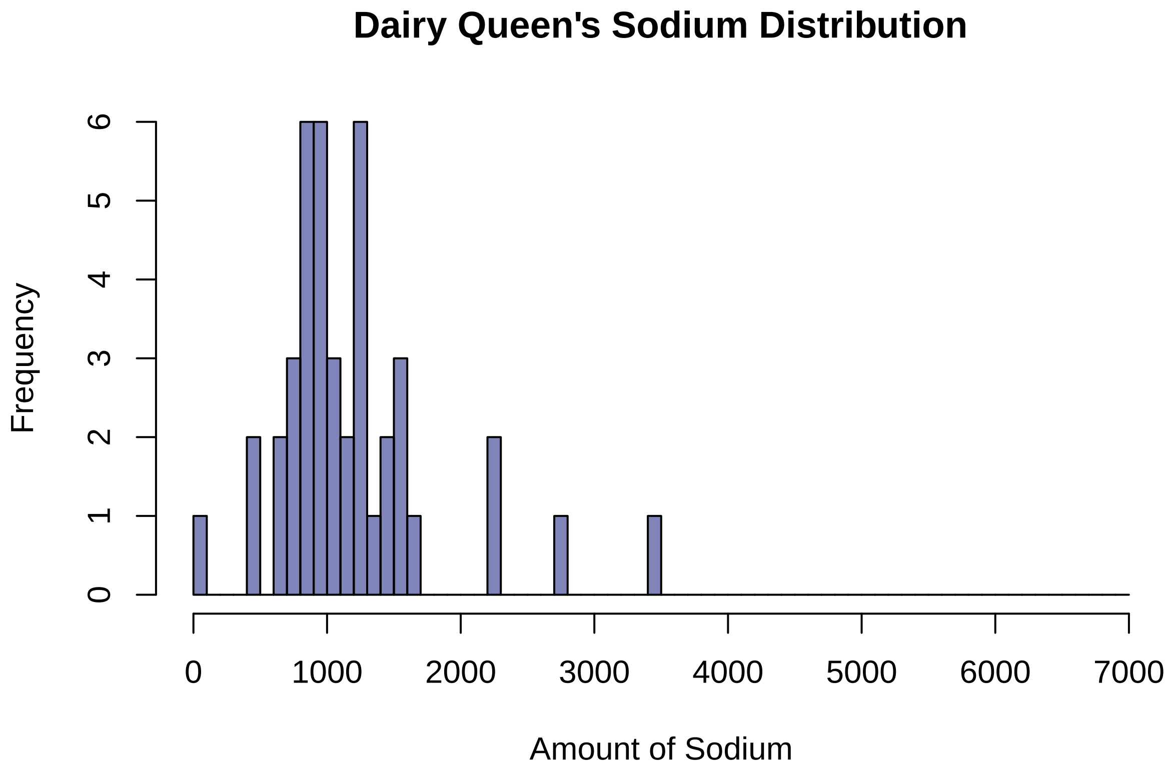
```
# although some were already computed, here they all are for convenience
mcdonalds <- fastfood %>%
  filter(restaurant == "Mcdonalds")
dairy_queen <- fastfood %>%
  filter(restaurant == "Dairy Queen")
chick_fil_a <- fastfood %>%
  filter(restaurant == "Chick Fil-A")
sonic <- fastfood %>%
  filter(restaurant == "Sonic")
arbys <- fastfood %>%
  filter(restaurant == "Arbys")
burger_king <- fastfood %>%
  filter(restaurant == "Burger King")
subway <- fastfood %>%
  filter(restaurant == "Subway")
tacobell <- fastfood %>%
  filter(restaurant == "Taco Bell")
```

Thinking ahead, instead of running many analyses on all distributions, it would be useful to eliminate those restaurant distributions that are distinctly not normal. To determine which ones do not pass the possibility of a normal distribution, I used the following histograms.

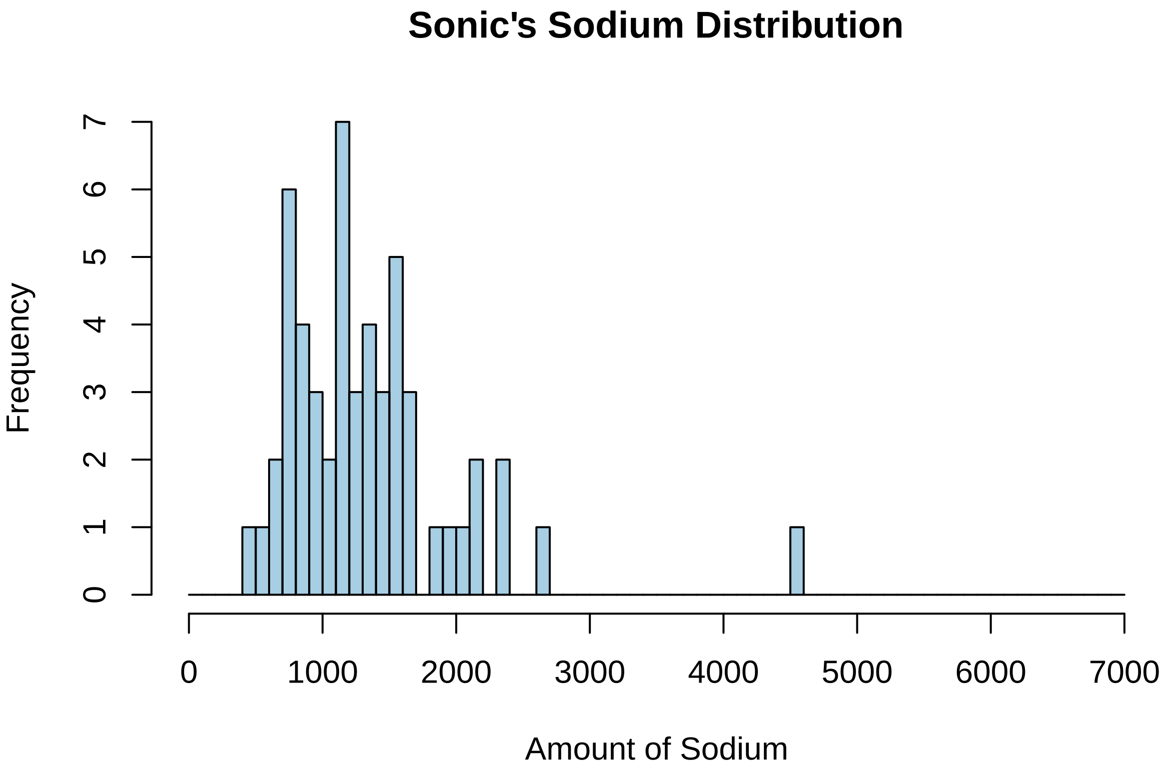
```
hist(mcdonalds$sodium,
     col = rgb(red = 0.80, green = 0.80, blue = 0, alpha = 0.50),
     xlab = "Amount of Sodium",
     main = "McDonald's Sodium Distribution",
     breaks = seq(0,7000,100))
```



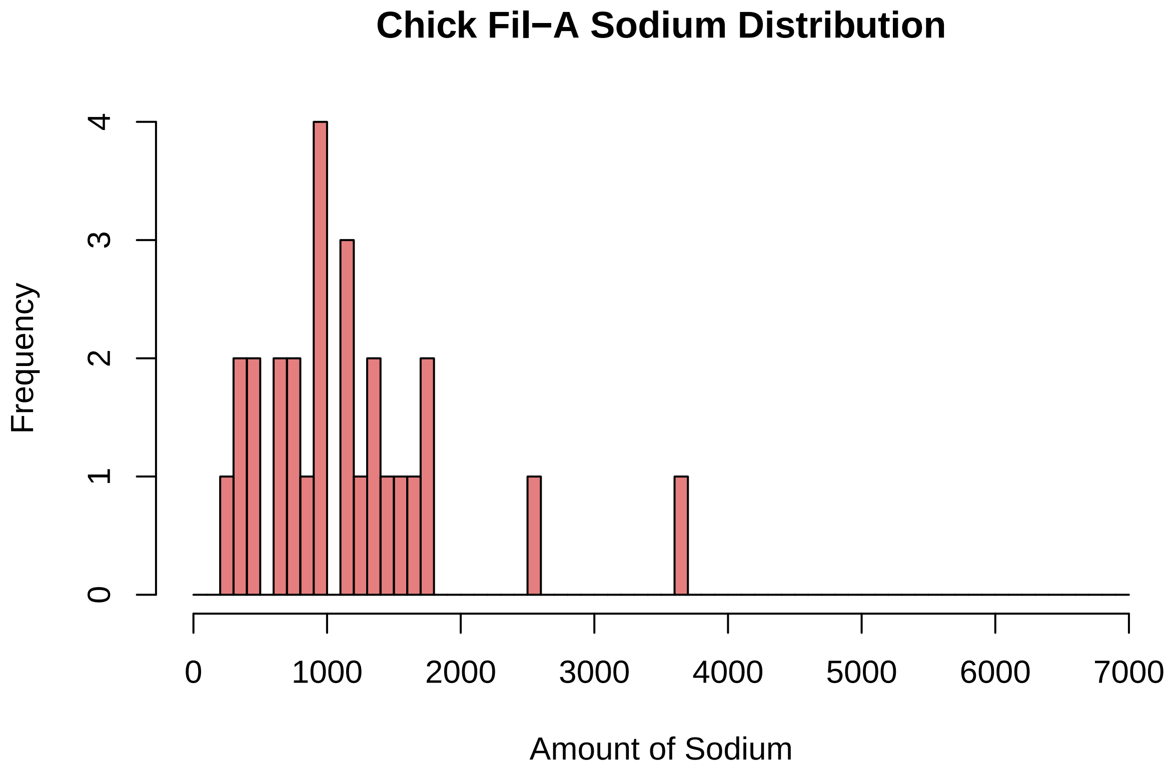
```r
hist(dairy_queen$sodium,
     col = rgb(red = 0, green = 0.05, blue = 0.45, alpha = 0.50),
     xlab = "Amount of Sodium",
     main = "Dairy Queen's Sodium Distribution",
     breaks = seq(0,7000,100))
```



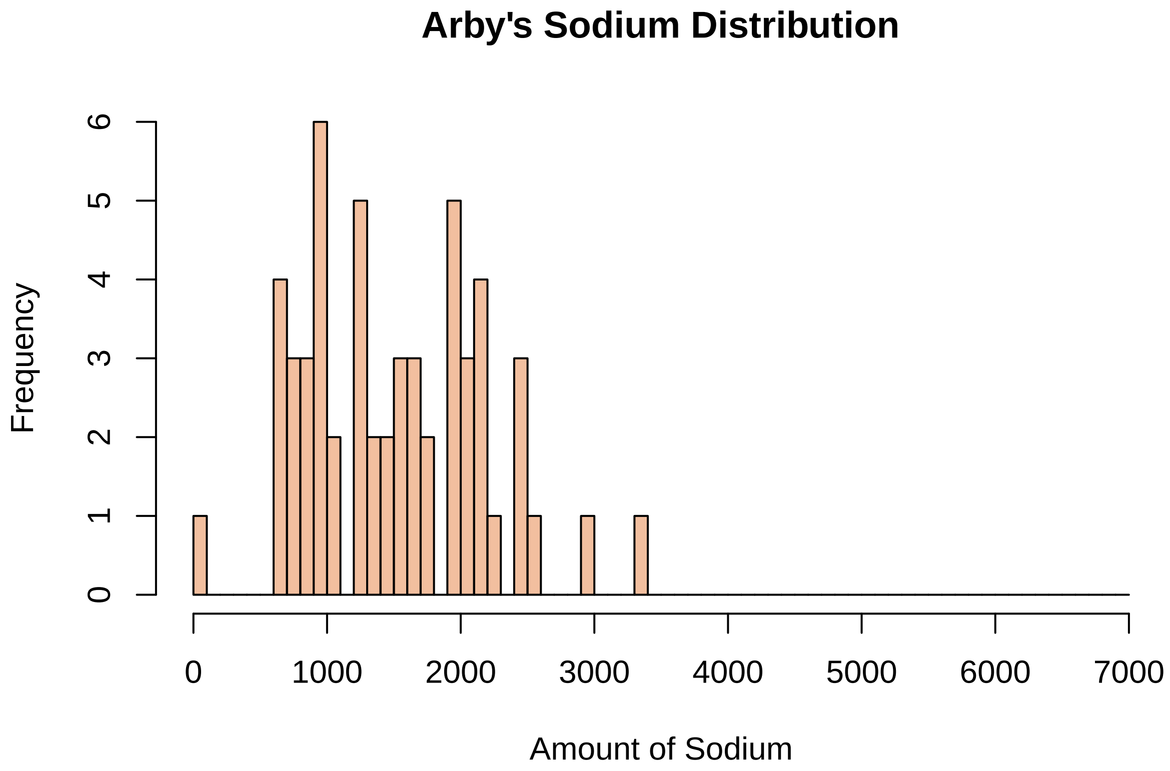
```
hist(sonic$sodium,
     col = rgb(red = 0.00, green = 0.45, blue = 0.69, alpha = 0.35),
     xlab = "Amount of Sodium",
     main = "Sonic’s Sodium Distribution",
     breaks = seq(0,7000,100))
```



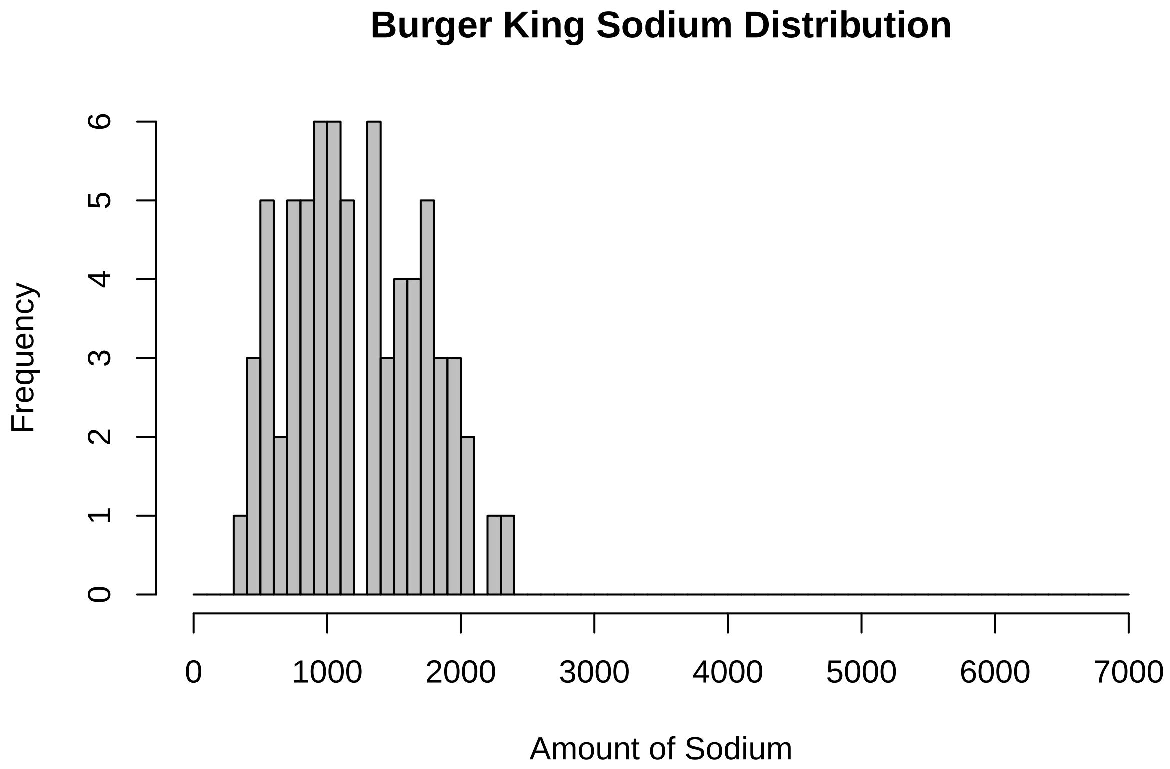
```r
hist(chick_fil_a$sodium,
     col = rgb(red = .80, green = 0.00, blue = 0.00, alpha = 0.50),
     xlab = "Amount of Sodium",
     main = "Chick Fil-A Sodium Distribution",
     breaks = seq(0,7000,100))
```



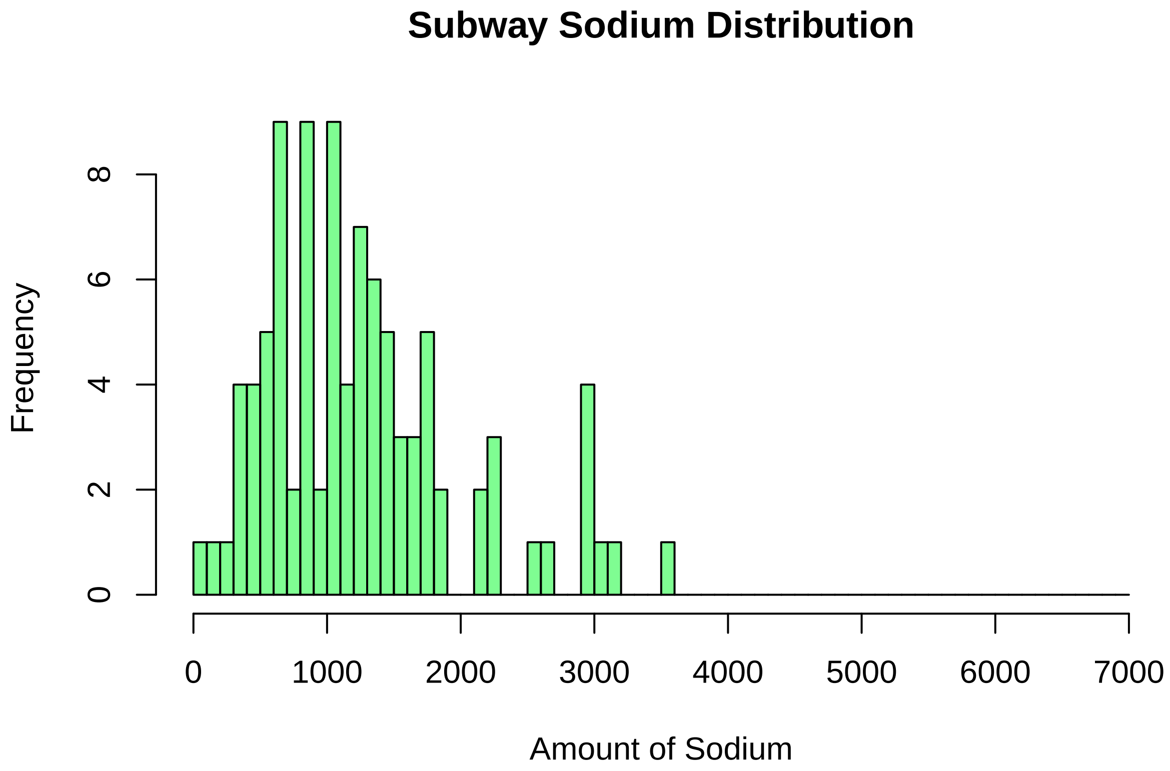
```
hist(arbys$sodium,
     col = rgb(red = 0.90, green = 0.50, blue = 0.25, alpha = 0.50),
     xlab = "Amount of Sodium",
     main = "Arby’s Sodium Distribution",
     breaks = seq(0,7000,100))
```



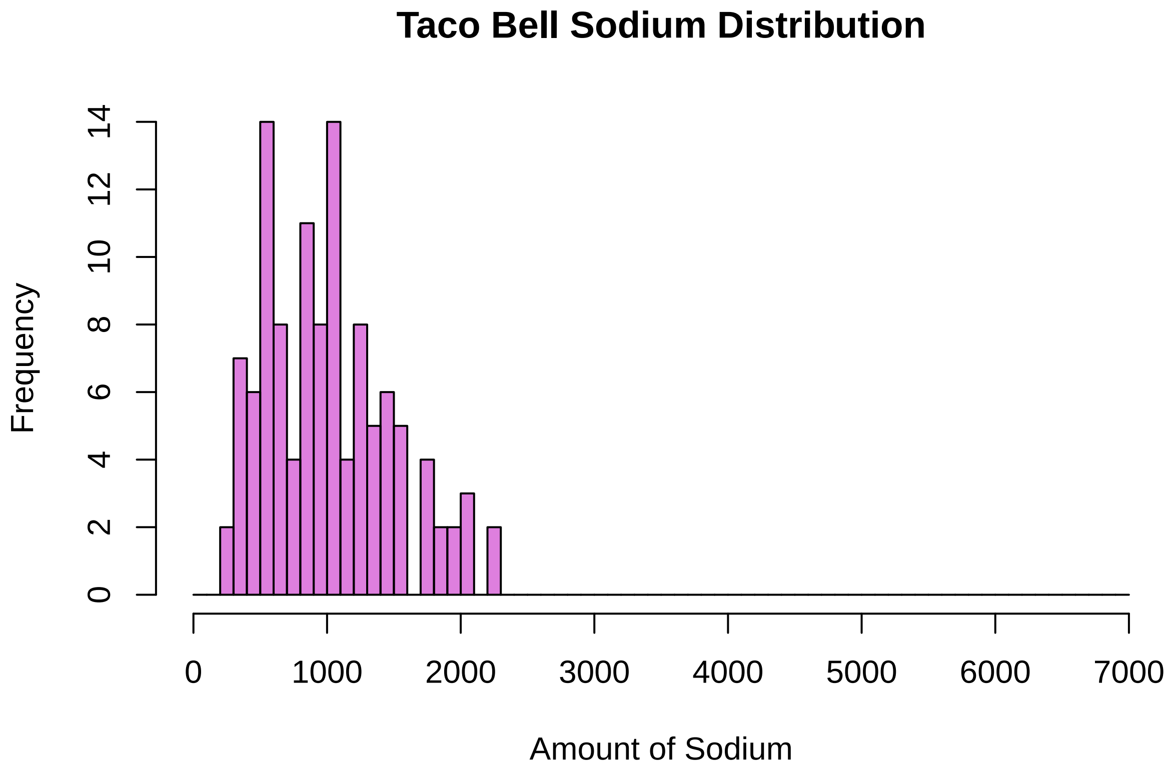
```r
hist(burger_king$sodium,
     col = rgb(red = 0.50, green = 0.50, blue = 0.50, alpha = 0.50),
     xlab = "Amount of Sodium",
     main = "Burger King Sodium Distribution",
     breaks = seq(0,7000,100))
```



```r
hist(subway$sodium,
     col = rgb(red = 0, green = 0.99, blue = 0.15, alpha = 0.50),
     xlab = "Amount of Sodium",
     main = "Subway Sodium Distribution",
     breaks = seq(0,7000,100))
```



```r
hist(tacobell$sodium,
     col = rgb(red = 0.75, green = 0.00, blue = 0.75, alpha = 0.50),
     xlab = "Amount of Sodium",
     main = "Taco Bell Sodium Distribution",
     breaks = seq(0,7000,100))
```



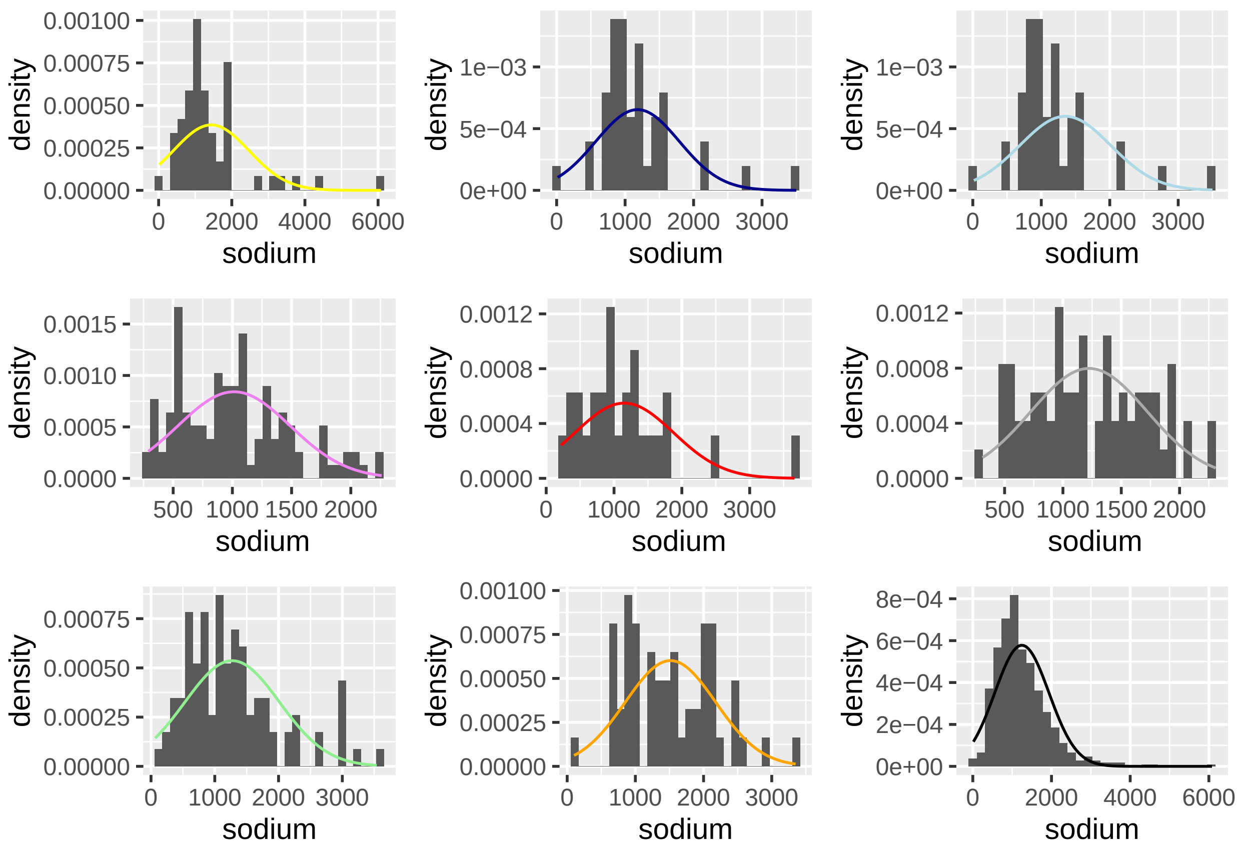
Given that a normal distribution is an unimodal, symmetric distribution, it should create a generally bell-shaped graph centered at the mean (and median). Thinking about this as a guide we can compare.

Based on the histograms of each restaurant, there are a few distributions that stand out. Arbys appears to have two modes, and thus is not a normal distribution. McDonalds it not symmetric about the mean since its shape stretches towards higher amounts of sodium and it may also be bimodal at a binwidth of 100. This should be confirmed in the density plots below.

```
###################
# McDonalds Density Distribution
mcd_sod_mean <- mean(mcdonalds$sodium)
mcd_sod_sd   <- sd(mcdonalds$sodium)
mcd_gghist <- ggplot(data = mcdonalds, aes(x = sodium)) +
        geom_blank() +
        geom_histogram(aes(y = ..density..)) +
        stat_function(fun = dnorm, args = c(mean = mcd_sod_mean, sd = mcd_sod_sd), col = "yellow")

###################
# Dairy Queen Density Distribution
dq_sod_mean <- mean(dairy_queen$sodium)
dq_sod_sd   <- sd(dairy_queen$sodium)
dq_gghist <- ggplot(data = dairy_queen, aes(x = sodium)) +
        geom_blank() +
        geom_histogram(aes(y = ..density..)) +
        stat_function(fun = dnorm, args = c(mean = dq_sod_mean, sd = dq_sod_sd), col = "dark blue")

###################
```

```r
# Sonic Density Distribution
sonic_sod_mean <- mean(sonic$sodium)
sonic_sod_sd   <- sd(sonic$sodium)
sonic_gghist <- ggplot(data = dairy_queen, aes(x = sodium)) +
        geom_blank() +
        geom_histogram(aes(y = ..density..)) +
        stat_function(fun = dnorm, args = c(mean = sonic_sod_mean, sd = sonic_sod_sd), col = "light blu

###################
# Arbys Density Distribution
arb_sod_mean <- mean(arbys$sodium)
arb_sod_sd   <- sd(arbys$sodium)
arb_gghist <- ggplot(data = arbys, aes(x = sodium)) +
        geom_blank() +
        geom_histogram(aes(y = ..density..)) +
        stat_function(fun = dnorm, args = c(mean = arb_sod_mean, sd = arb_sod_sd), col = "orange")


###################
# Subway Density Distribution
sub_sod_mean <- mean(subway$sodium)
sub_sod_sd   <- sd(subway$sodium)
sub_gghist <- ggplot(data = subway, aes(x = sodium)) +
        geom_blank() +
        geom_histogram(aes(y = ..density..)) +
        stat_function(fun = dnorm, args = c(mean = sub_sod_mean, sd = sub_sod_sd), col = "light green")

###################
# Burger King Density Distribution
bk_sod_mean <- mean(burger_king$sodium)
bk_sod_sd   <- sd(burger_king$sodium)
bk_gghist <- ggplot(data = burger_king, aes(x = sodium)) +
        geom_blank() +
        geom_histogram(aes(y = ..density..)) +
        stat_function(fun = dnorm, args = c(mean = bk_sod_mean, sd = bk_sod_sd), col = "dark grey")

###################
# Chick Fil-A Density Distribution
cfa_sod_mean <- mean(chick_fil_a$sodium)
cfa_sod_sd   <- sd(chick_fil_a$sodium)
cfa_gghist <- ggplot(data = chick_fil_a, aes(x = sodium)) +
        geom_blank() +
        geom_histogram(aes(y = ..density..)) +
        stat_function(fun = dnorm, args = c(mean = cfa_sod_mean, sd = cfa_sod_sd), col = "red")

###################
# Taco Bell Density Distribution
tb_sod_mean <- mean(tacobell$sodium)
tb_sod_sd   <- sd(tacobell$sodium)
tb_gghist <- ggplot(data = tacobell, aes(x = sodium)) +
        geom_blank() +
        geom_histogram(aes(y = ..density..)) +
        stat_function(fun = dnorm, args = c(mean = tb_sod_mean, sd = tb_sod_sd), col = "violet")
```

```
###################
# Fast Food Density Distribution
all_sod_mean <- mean(fastfood$sodium)
all_sod_sd   <- sd(fastfood$sodium)
all_gghist <- ggplot(data = fastfood, aes(x = sodium)) +
        geom_blank() +
        geom_histogram(aes(y = ..density..)) +
        stat_function(fun = dnorm, args = c(mean = all_sod_mean, sd = all_sod_sd), col = "black")

# Bring it all together
ggarrange(mcd_gghist, dq_gghist, sonic_gghist, tb_gghist, cfa_gghist, bk_gghist, sub_gghist, arb_gghist
```



Here, a normalized line is plotted on top of the histogram of each restaurant's distribution of sodium amounts. In Orange, we can see that there should be a mean around 1500 where the line builds up into a mound-like shape. The histogram of the data, however, does not follow this trend, confirming the exclusion of Arbys from being considered a normal distribution.

McDonalds too, should be excluded because its not symmetric enough to be a normal distribution. It has a long tail of values extending its shape to the right, much more than any other histogram. The rest will remain for further review.

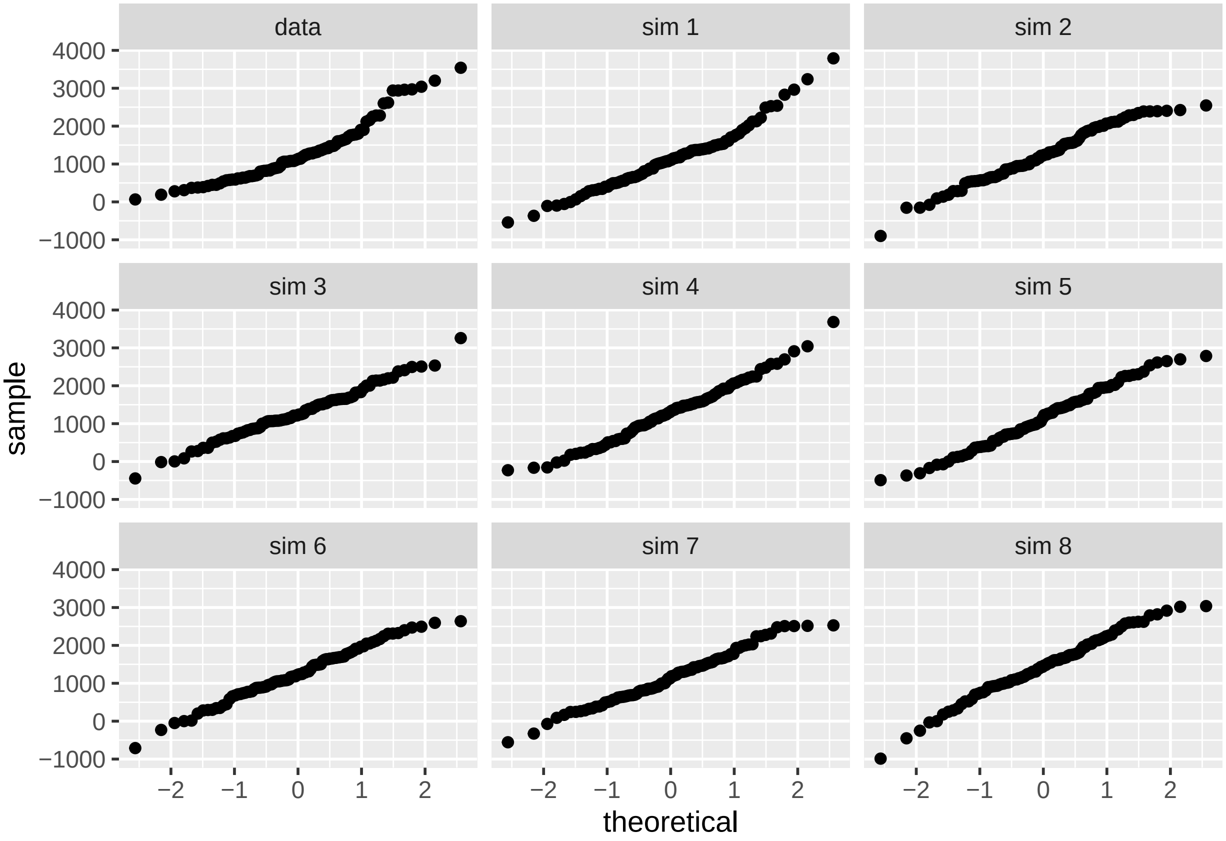
This leaves us with these restaurants:

- Subway
- Chick Fil-A
- Dairy Queen

- Burger King
- Taco Bell
- Sonic

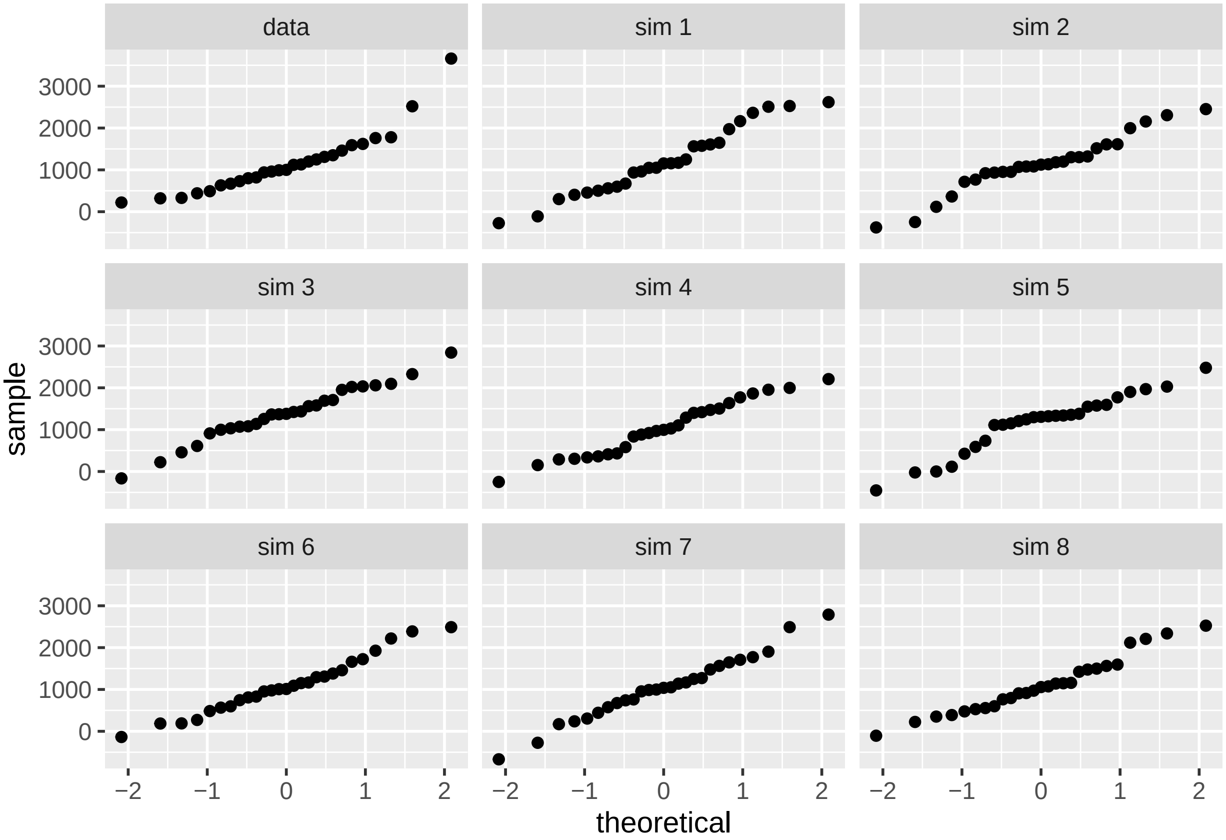
For review using qq plots against simulated normal distributions. Subway is first.

```r
# QQ Plots for Distribution of Sodium by Subway
set.seed(512551)
qqnormsim(sample = sodium, data = subway)
```



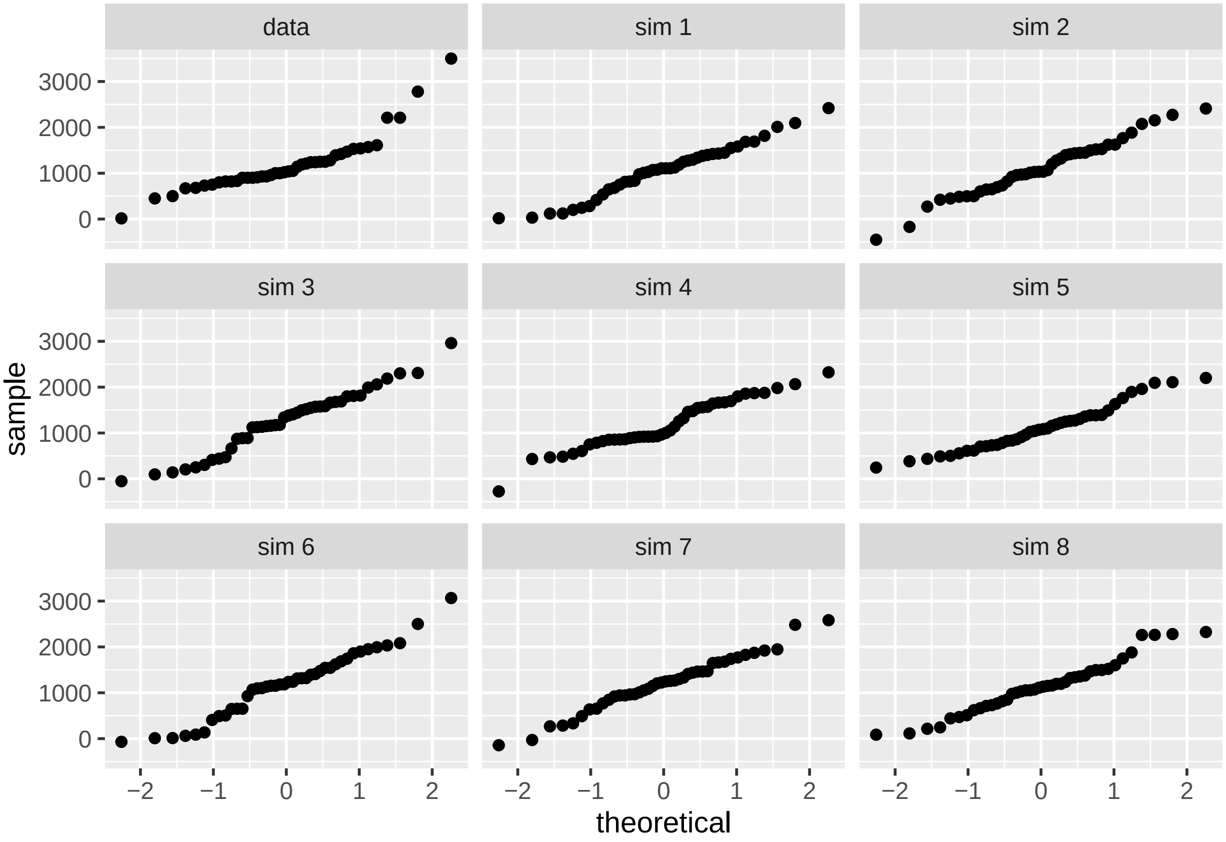
This distribution is very close to normal but it could be better. The simulations show a straight diagonal line while the upper right portion of the data appears to jump and flatten out slightly.

```r
# QQ Plots for Distribution of Sodium by Chick Fil-A
set.seed(5125512)
qqnormsim(sample = sodium, data = chick_fil_a)
```



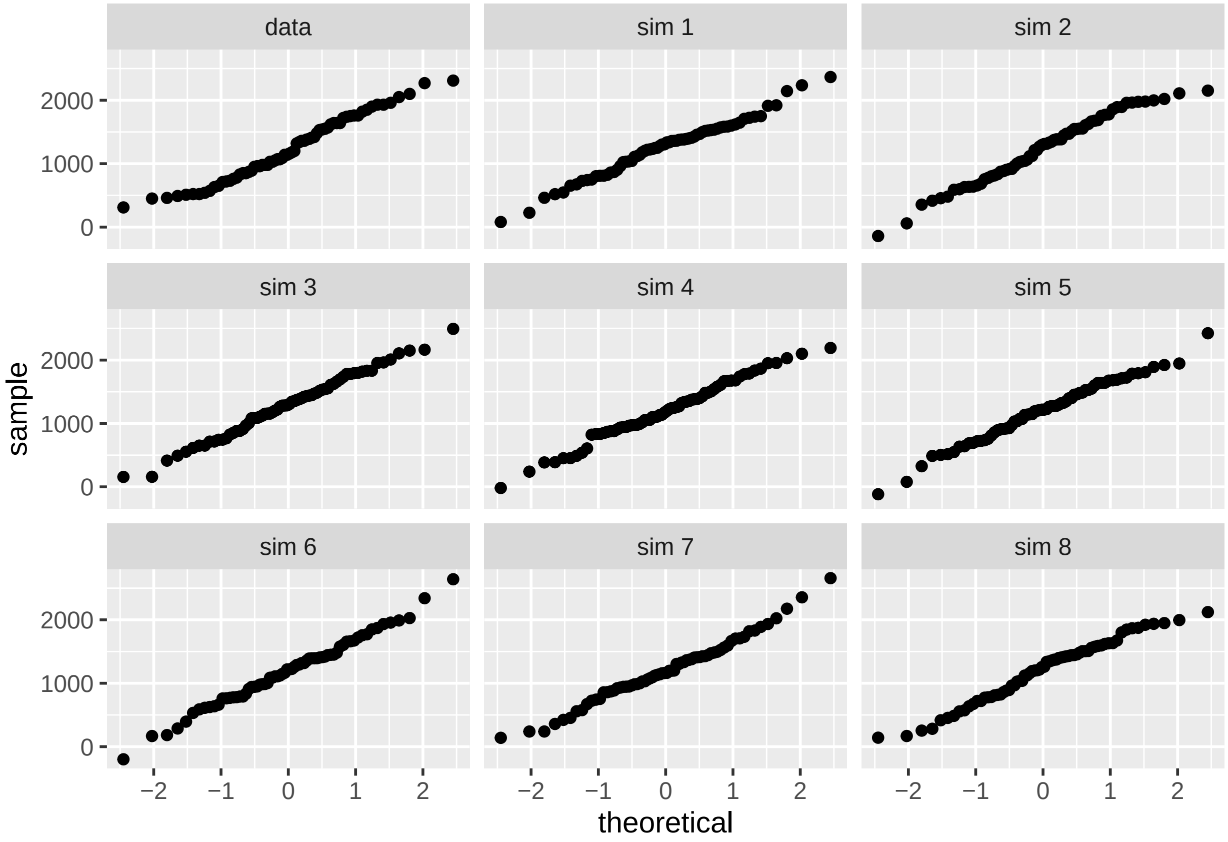
Due to the its deviation from normal, this distribution of Chick Fil-A's sodium is worse (at being normal) than subways.

```r
# QQ Plots for Distribution of Sodium by Dairy Queen
set.seed(51255123)
qqnormsim(sample = sodium, data = dairy_queen)
```



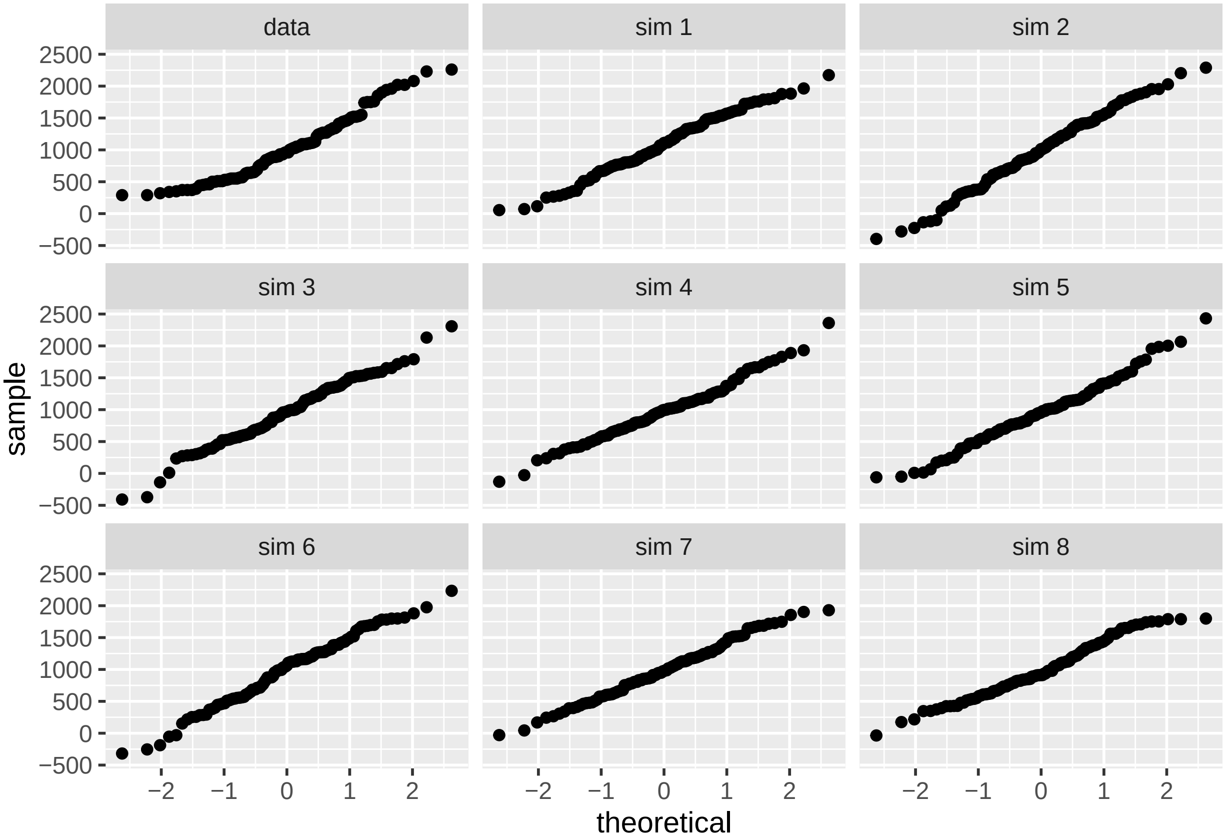
Dairy Queen's sodium distribution is less normal than Chick Fil-A's and also jumps from one group of values to the next.

```
# QQ Plots for Distribution of Sodium by Burger King
set.seed(512551234)
qqnormsim(sample = sodium, data = burger_king)
```



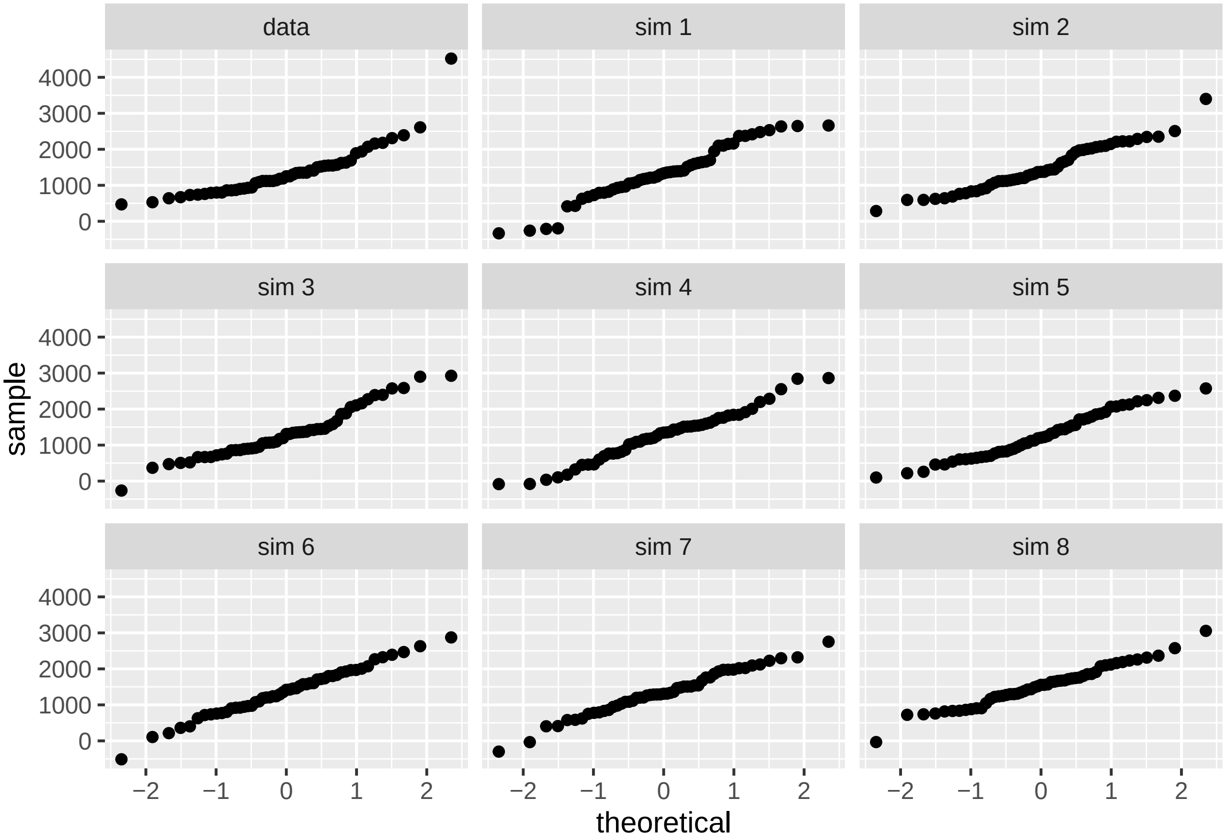
Burger King appears to follow a relatively normal pattern. It does, however, contain one small jump in the data in the middle of its distribution. Otherwise, it appears very similar to the other normal simulations.

```r
# QQ Plots for Distribution of Sodium by Taco Bell
set.seed(51255)
qqnormsim(sample = sodium, data = tacobell)
```



Taco bell also has a normal distribution but it too contains a jump in values. In this case, it is towards the end of its distribution. Otherwise, it is very similar to the rest of the simulations.

```
# QQ Plots for Distribution of Sodium by Sonic
set.seed(55123456)
qqnormsim(sample = sodium, data = sonic)
```



Sonic is not aligned with a normal distribution as the simulations each appear to be. Though it seems due to a group of outliers in the data.

Of these distributions, it appears Taco Bell or Burger King has the most normal distribution. Selecting the range of data from those, we may be able to narrow down which one has a closer to normal distribution by comparing empirical calculations of its data to theoretical normal.

```
# Taco bell empirical to theoretical comparison less than mean
pnorm(q = tb_sod_mean, mean = tb_sod_mean, sd = tb_sod_sd)
```

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

```
tacobell %>%
  filter(sodium < tb_sod_mean) %>%
  summarise(percent = n() / nrow(tacobell))
```

```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1   0.530
```

```
# Taco bell empirical to theoretical comparison greater than mean
1 - pnorm(q = tb_sod_mean, mean = tb_sod_mean, sd = tb_sod_sd)
```

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

```
tacobell %>%
  filter(sodium > tb_sod_mean) %>%
  summarise(percent = n() / nrow(tacobell))
```

```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1   0.470
```

Rather picking numbers at random or trying to compare the entire distribution at once, the calculated and theoretical distributions to the left and right of the mean were used. Thus, for the theoretical normal distributions, the value should always be 0.50 (or 50%) because the mean values occurs where half of the data is distributed above, and half below. For this reason, it should be easier to compare both distributions fairly. For example, Taco Bell's distribution was calculated to have 0.53043 data to the left of, or less than the mean, and 0.46957 to right right of, or greater than the mean. Perfectly normal distributions (our theoretical) by nature have 0.5 of their data above and below the mean. To find how far Taco Bell's distribution was, we calculate the absolute value of the difference between expected and reality. This difference is called the deviation from normal as shown here;

$$Abs(Theorical - Reality) = Deviation$$

For Taco Bell these deviations are 0.03043 where data is less than the mean, and 0.03043 where data is greater than the mean.

```
# Burger King empirical to theoretical comparison less than mean
pnorm(q = bk_sod_mean, mean = bk_sod_mean, sd = bk_sod_sd)
```

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

```
burger_king %>%
  filter(sodium < bk_sod_mean) %>%
  summarise(percent = n() / nrow(burger_king))
```

```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1   0.543
```

```
# Burger King empirical to theoretical comparison greater than mean
1 - pnorm(q = bk_sod_mean, mean = bk_sod_mean, sd = bk_sod_sd)
```

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

```
burger_king %>%
  filter(sodium > bk_sod_mean) %>%
  summarise(percent = n() / nrow(burger_king))
```

```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1   0.457
```

For Burger King these deviations are 0.04286 where data is less than the mean, and 0.04286 where data is greater than the mean. To finish, we can sum the two differences for each restaurant and compare which one is greater. The greater one will indicate the on that deviates farther from theoretical normal than the other.

To put these into a table we have;

Restaurant	Theoretical	Empirical	Deviation	Total
Taco Bell >	0.50	0.46957	0.03043	-
Taco Bell <	0.50	0.53043	0.03043	0.06086
Burger King >	0.50	0.45714	0.04286	-
Burger King <	0.50	0.54286	0.04286	0.08572
Total	2.00	2.00	0.14658	-

Where the greater or less than symbols indicate whether it was greater or less than the mean of the restaurant.

```r
# Difference in deviation from normal
abs((0.04286*2)-(0.03043*2))
```

```
## [1] 0.02486
```

Using the data from the table and comparing their deviations from normal, it shows Taco Bell is the more normal of the two distributions. Having a total deviation from the mean of 0.06086 to Burger King's total deviation of 0.08572, Taco Bell is closer to agreement with a theoretical normal distribution by a margin of 0.02486.

Exercise 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?

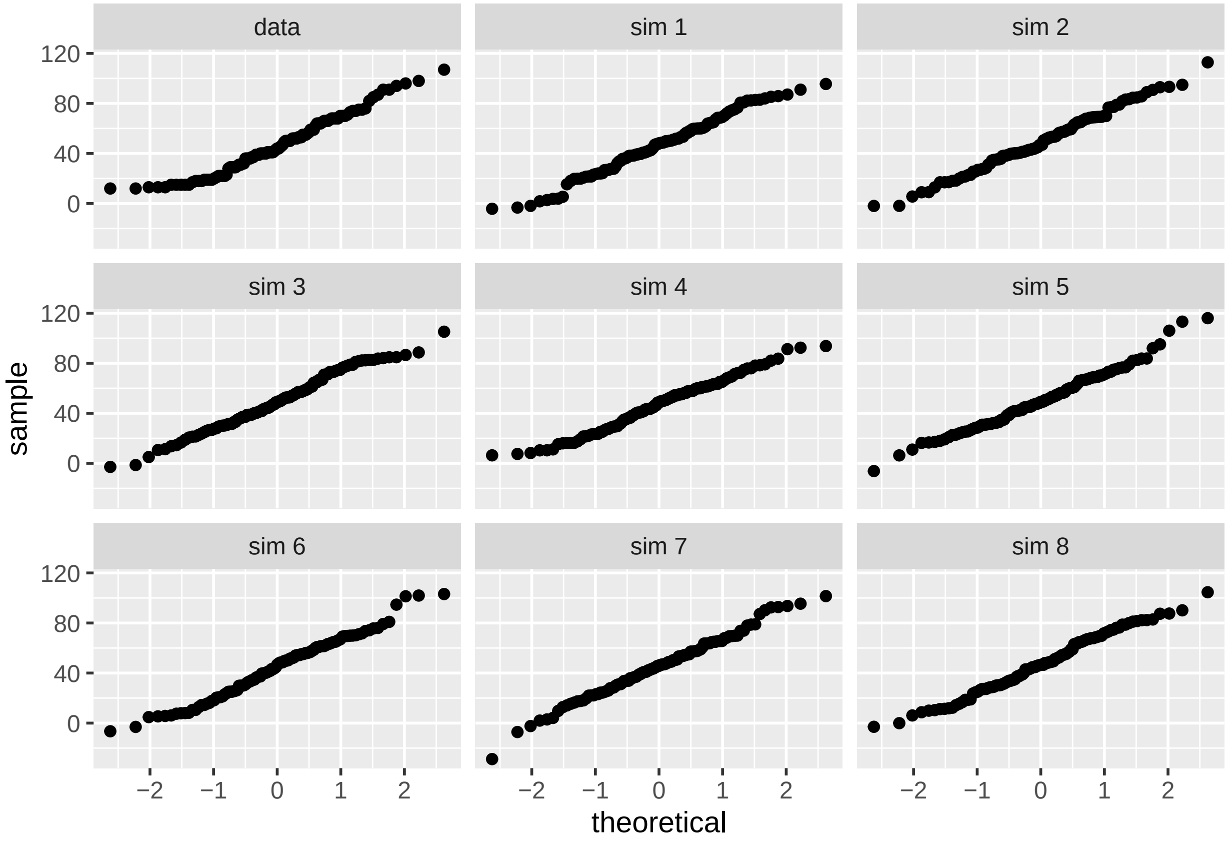
Stepwise distributions might come from the inherent grouping of menu items which are ultimately due to customer preferences. For example, the items causing the range of sodium content to be stepwise could be the menus including a range of salads to types of chicken-based meals, cured meats, or french fries. Each of which have their own range of sodium and do not gradually increase or decrease in sodium content. Fries have higher sodium content (based on how they are cooked - with salt) while salads should have hardly any sodium by comparison. A distinct jump should be seen if these were the only two options at the restaurant. The distribution shows this pattern over all groups too, which may indicate that restaurant menus contain limited options for food with sodium content between these groupings. In my opinion, is why some of the data appears to have a stepwise pattern.

Exercise 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.

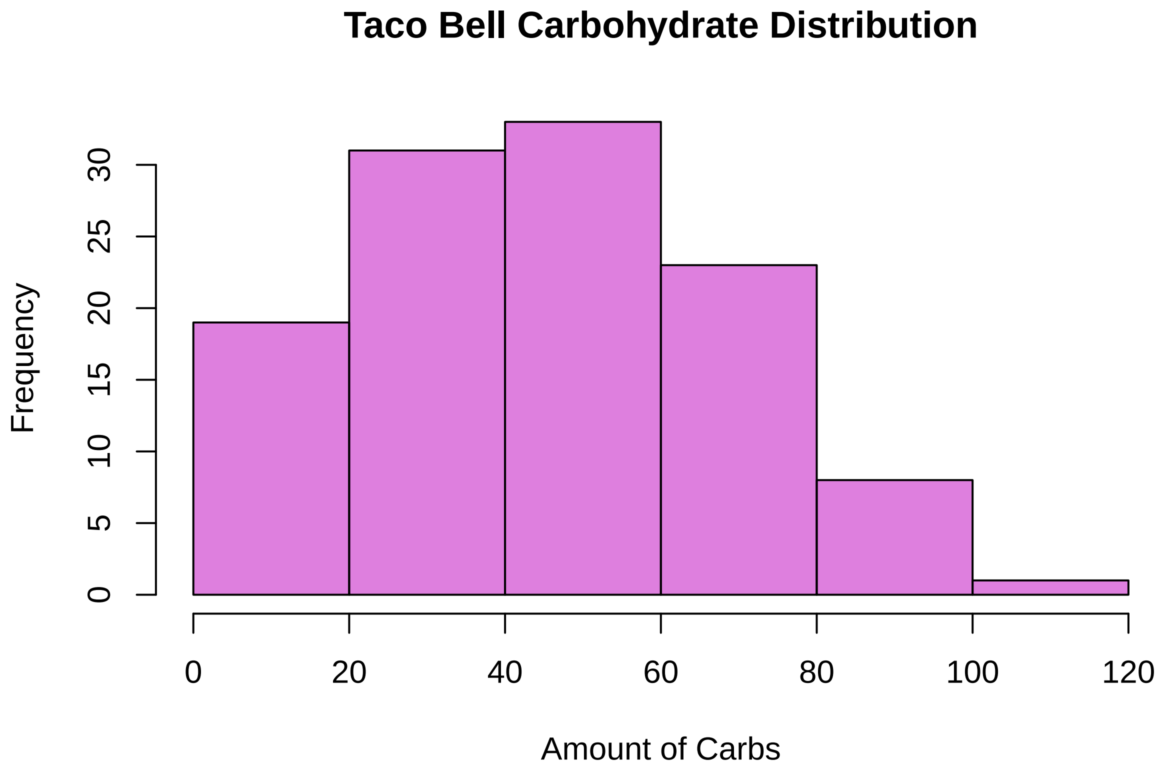
Taco bell is my choice and this is a normal probability plot for its distribution of total carbohydrates.

```
set.seed(12354)
qqnormsim(sample = total_carb, data = tacobell)
```



here is a histogram of its distribution.

```
hist(tacobell$total_carb,
     col = rgb(red = 0.75, green = 0.00, blue = 0.75, alpha = 0.50),
     xlab = "Amount of Carbs",
     main = "Taco Bell Carbohydrate Distribution",
     breaks = seq(0,120,20))
```



Based on this histogram and normality plot, the data is very close to a symmetric distribution. Though it does have a slightly higher concentration of values at higher levels of carbohydrates, it relatively balanced across its distribution. We can also compare its mean and median.

```
mean(tacobell$total_carb)
```

```
## [1] 46.63478
```

```
median(tacobell$total_carb)
```

```
## [1] 44
```

With the mean and median only differing by a small amount, this data is very close to a symmetric distribution.

. . .