The normal distribution

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)
```

Creating a reproducible lab report

To create your new lab report, in RStudio, go to New File -> R Markdown... Then, choose From Template and then choose `Lab Report for OpenIntro Statistics Labs` from the list of templates.

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)
head(fastfood,50)
```

```
## # A tibble: 50 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
##  7 Mcdonalds  Chee~      300     100        12       5       0.5          40
```

```
##  8 Mcdonalds  Clas~      510     210        24       4       0            65
##  9 Mcdonalds  Doub~      430     190        21      11       1            85
## 10 Mcdonalds  Doub~      770     400        45      21       2.5         175
## # ... with 40 more rows, and 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"))
```

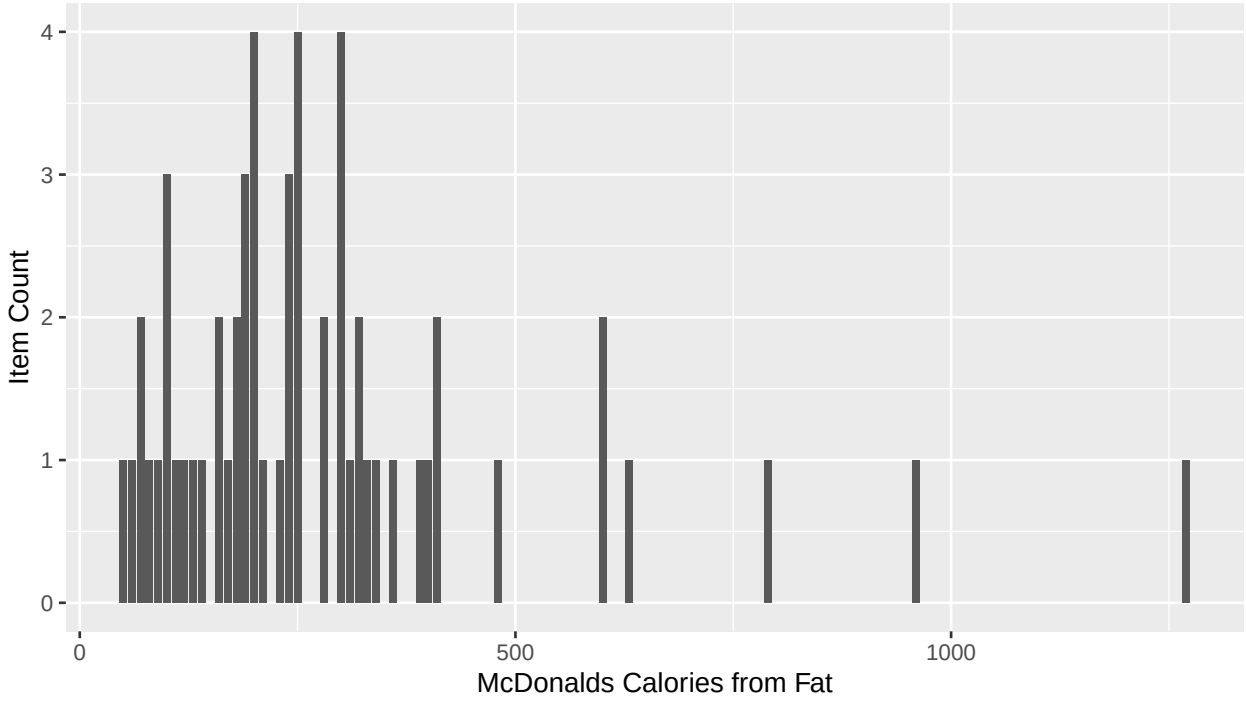
```
## # A tibble: 57 x 17
##    restaurant item  calories cal_fat total_fat sat_fat trans_fat cholesterol
##    <chr>      <chr>    <dbl>   <dbl>     <dbl>   <dbl>     <dbl>       <dbl>
##  1 Mcdonalds  Arti~      380      60         7       2       0            95
##  2 Mcdonalds  Sing~      840     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
##  7 Mcdonalds  Chee~      300     100        12       5       0.5          40
##  8 Mcdonalds  Clas~      510     210        24       4       0            65
##  9 Mcdonalds  Doub~      430     190        21      11       1            85
## 10 Mcdonalds  Doub~      770     400        45      21       2.5         175
## # ... with 47 more rows, and 9 more variables: sodium <dbl>, total_carb <dbl>,
## #   fiber <dbl>, sugar <dbl>, protein <dbl>, vit_a <dbl>, vit_c <dbl>,
## #   calcium <dbl>, salad <chr>
```

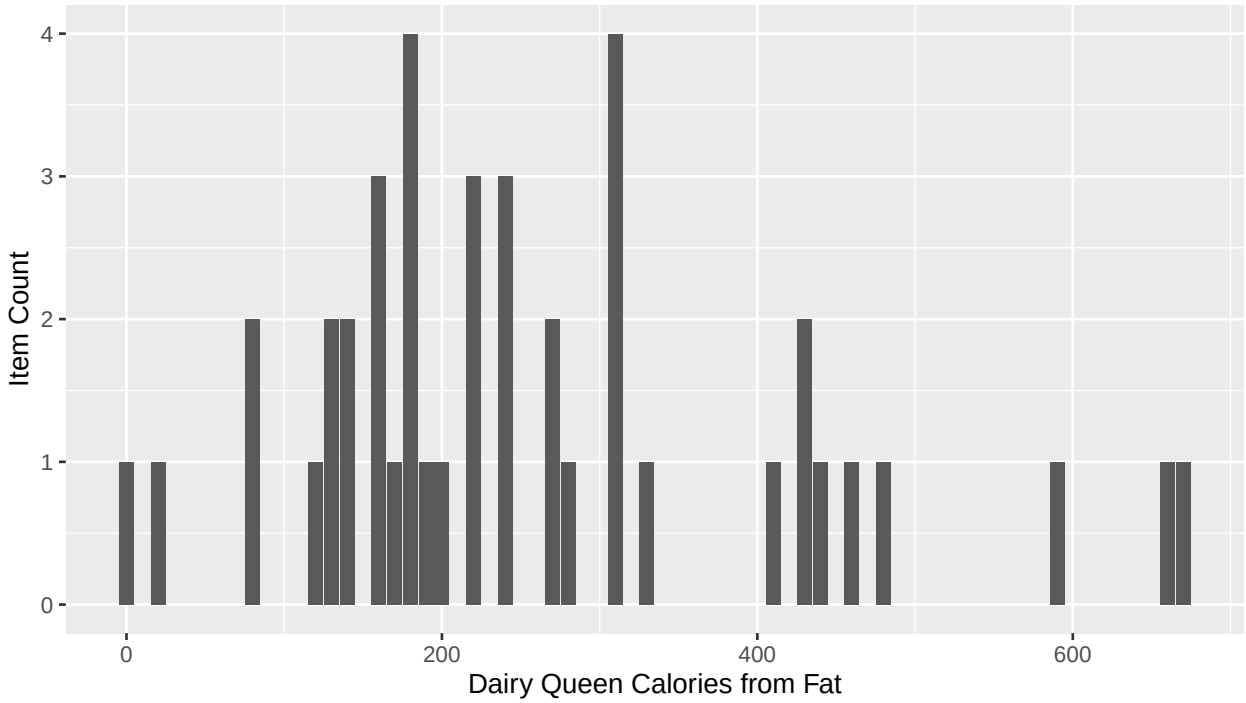
```
(dairy_queen <- fastfood %>%
  filter(restaurant == "Dairy Queen"))
```

```
## # A tibble: 42 x 17
##    restaurant item  calories cal_fat total_fat sat_fat trans_fat cholesterol
##    <chr>      <chr>    <dbl>   <dbl>     <dbl>   <dbl>     <dbl>       <dbl>
##  1 Dairy Que~ 1/2 ~     1000     660        74      26         2          170
##  2 Dairy Que~ 1/2 ~      800     460        51      20         2          135
##  3 Dairy Que~ 1/4 ~      630     330        37      13         1           95
##  4 Dairy Que~ 1/4 ~      540     270        30      11         1           70
##  5 Dairy Que~ 1/4 ~      570     310        35      11         1           75
##  6 Dairy Que~ Orig~      400     160        18       9         1           65
##  7 Dairy Que~ Orig~      630     310        34      18         2          125
##  8 Dairy Que~ 4 Pi~     1030     480        53       9         1           80
##  9 Dairy Que~ 6 Pi~     1260     590        66      11         1          120
## 10 Dairy Que~ Baco~      420     240        26      11         1           60
## # ... with 32 more rows, and 9 more variables: sodium <dbl>, total_carb <dbl>,
## #   fiber <dbl>, sugar <dbl>, protein <dbl>, vit_a <dbl>, vit_c <dbl>,
## #   calcium <dbl>, salad <chr>
```

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?

Answer: Both the McDonalds and Dairy Queen distribution seem very similar and appear to be skewed to the right, McDonalds more than Dairy Queen. Their calories from fat counts are very close to identical especially in the 100 to 500 calorie range. McDonalds does have a few more menu items than 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.

```
(mcd.mean <- mean(mcdonalds$cal_fat))
```

```
## [1] 285.614
```

```
(mcd.sd <- sd(mcdonalds$cal_fat))
```

```
## [1] 220.8993
```

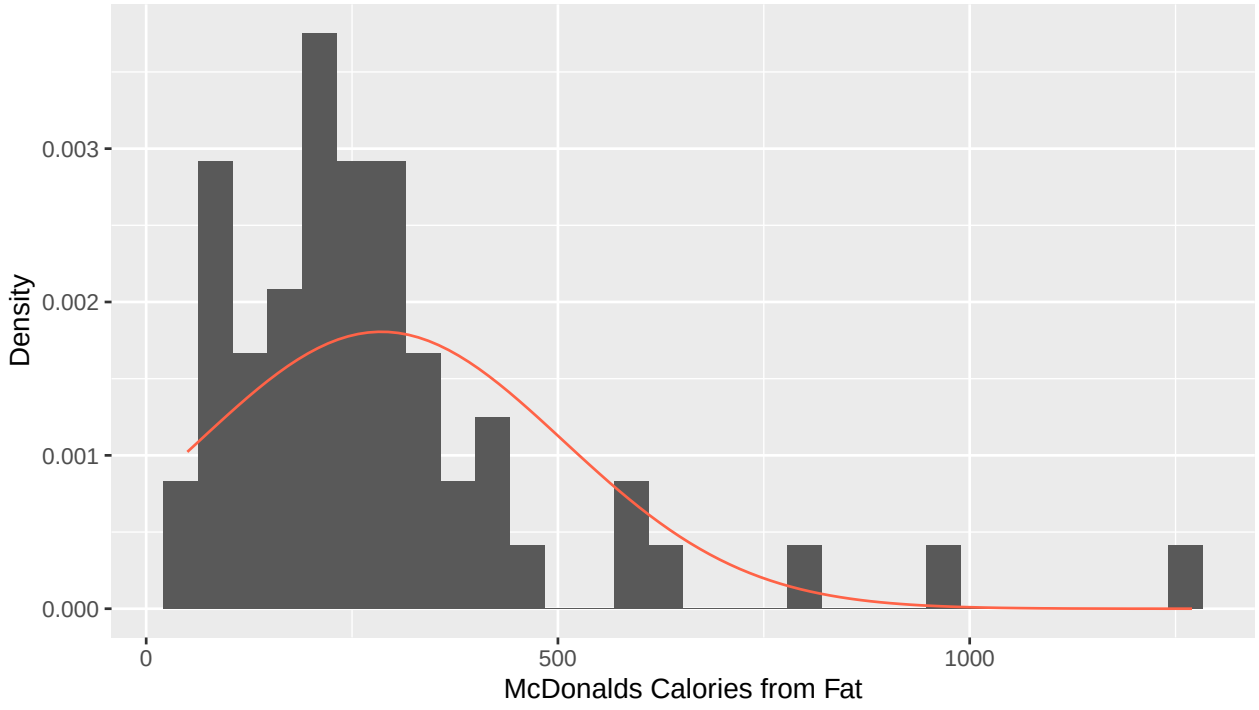
```
(dqmean <- mean(dairy_queen$cal_fat))
```

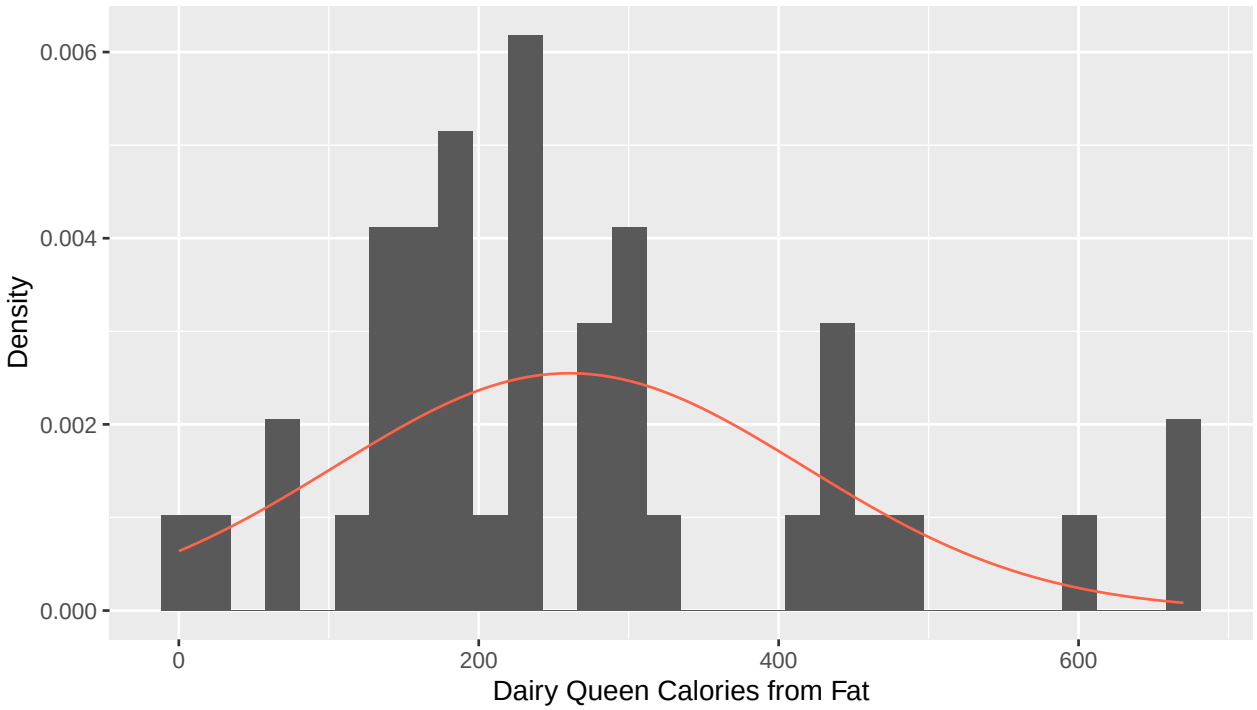
```
## [1] 260.4762
```

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

```
## [1] 156.4851
```

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.





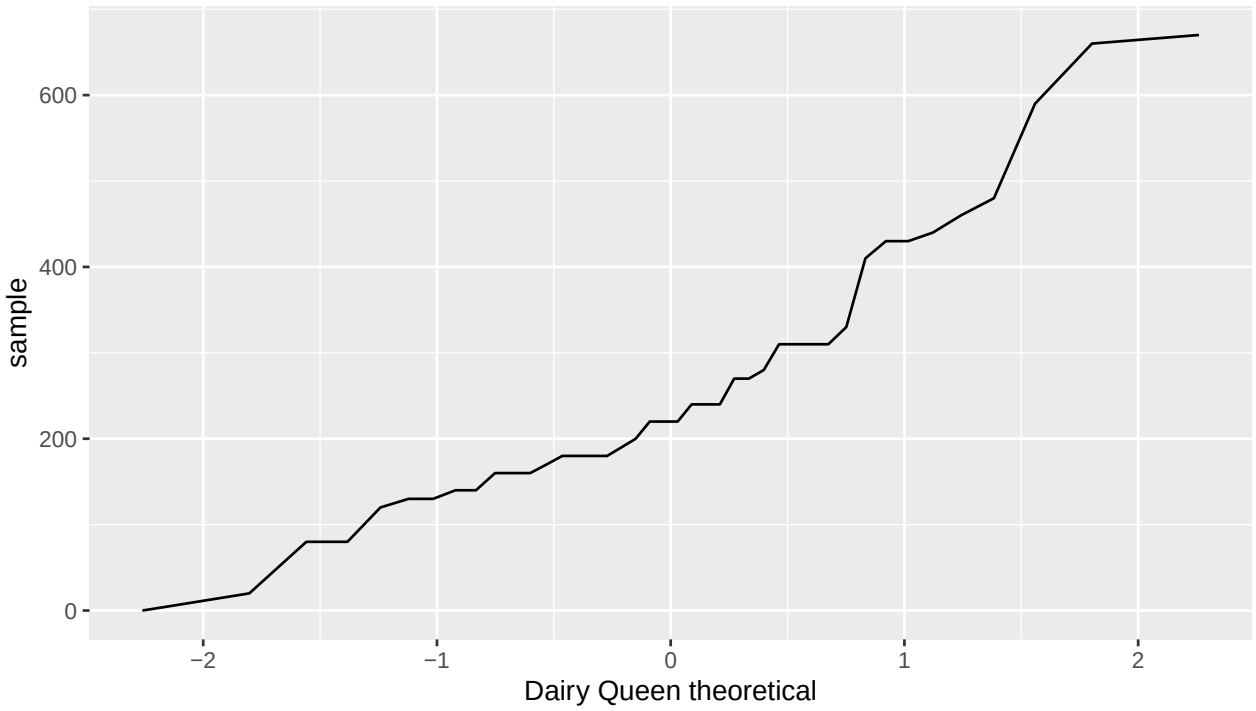
After initializing a blank plot with `geom_blank()`, the `ggplot2` package (within the `tidyverse`) allows us to add additional layers. The first layer is a density histogram. The second layer is a statistical function – the density of the normal curve, `dnorm`. We specify that we want the curve to have the same mean and standard deviation as the column of female heights. 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?

Answer: The Dairy Queen plot though having a slight right skew I would say that it aligns more closely to the normal distribution curve shown.

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".



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`.

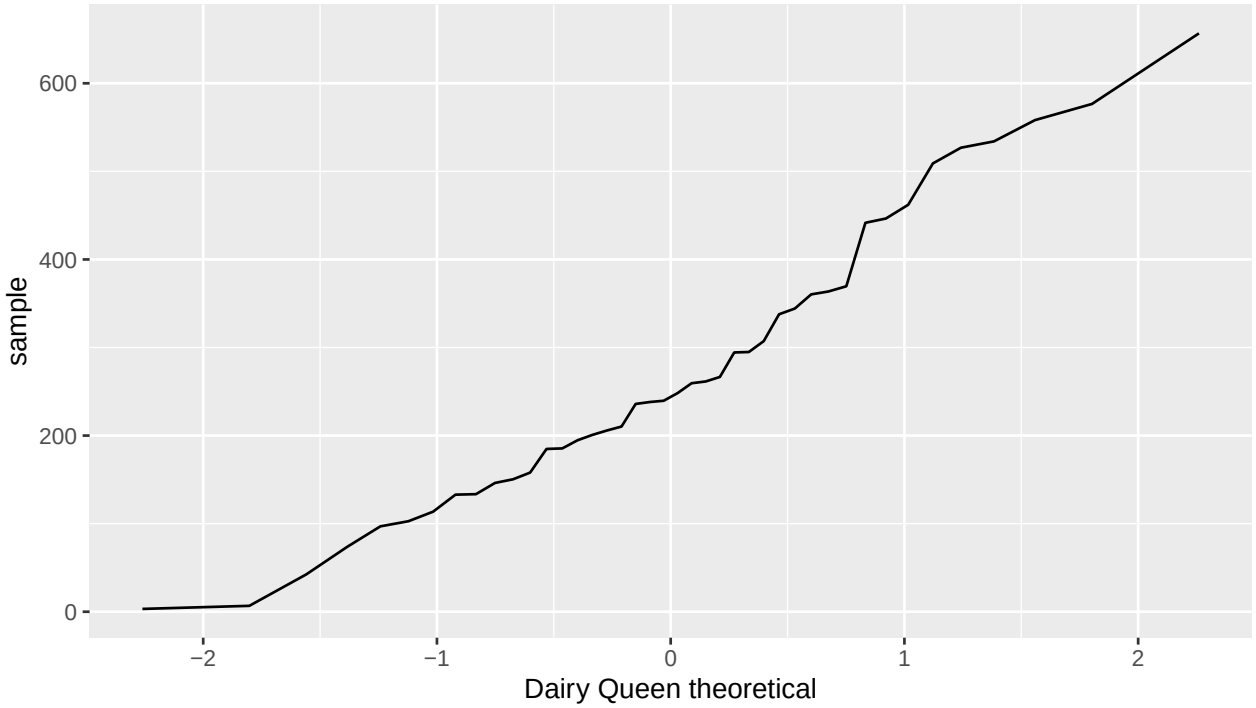
```
(sim_norm <- rnorm(n = nrow(dairy_queen), mean = dqmean, sd = dqsd))
```

```
##  [1] 337.751419 446.465590 294.385207 360.357140 235.931835 133.480494
##  [7]   6.702077 238.090519 307.284240 157.982278 184.836818 259.453190
## [13] 146.252351 534.022903 205.863525 248.284792 526.857353  96.967886
## [19] 266.538945 509.051956 261.438616 294.884747 210.255742 132.953529
## [25] 344.267850 150.373760 656.707345 576.569587 185.438360 441.603408
## [31] 200.821326  42.452529 462.056317 363.596874 369.511651 239.549462
## [37] 194.713398   3.207482 558.221807 113.641524 102.763599  73.858585
```

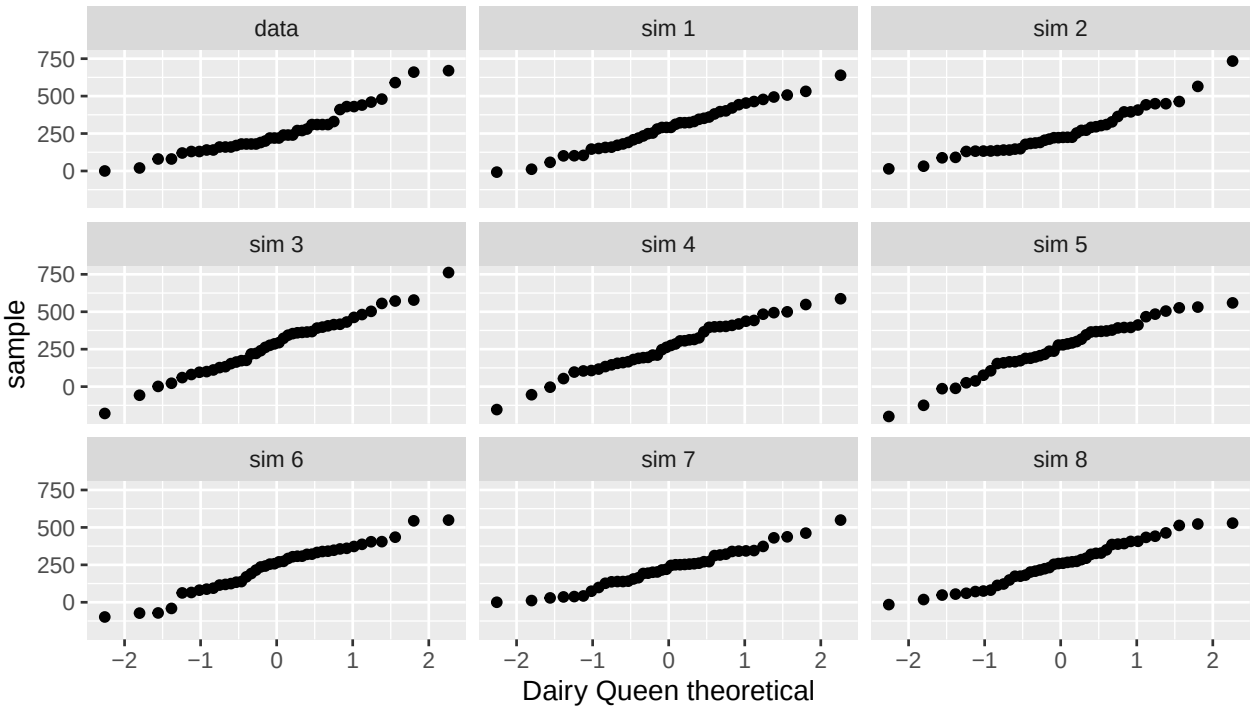
The first argument indicates how many numbers you'd like to generate, which we specify to be the same number of menu items in the `dairy_queen` data set using the `nrow()` function. The last two arguments determine the mean and standard deviation of the normal distribution from which the simulated sample 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 dataframe, it can be put directly into the `sample` argument and the `data` argument can be dropped.)

Answer: The Dairy Queen sim_norm normal probability plat is a little smoother it is almost identical to the actual normal plot. Both still show that the DQ distribution for fat calories is somewhat close to normal.



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.

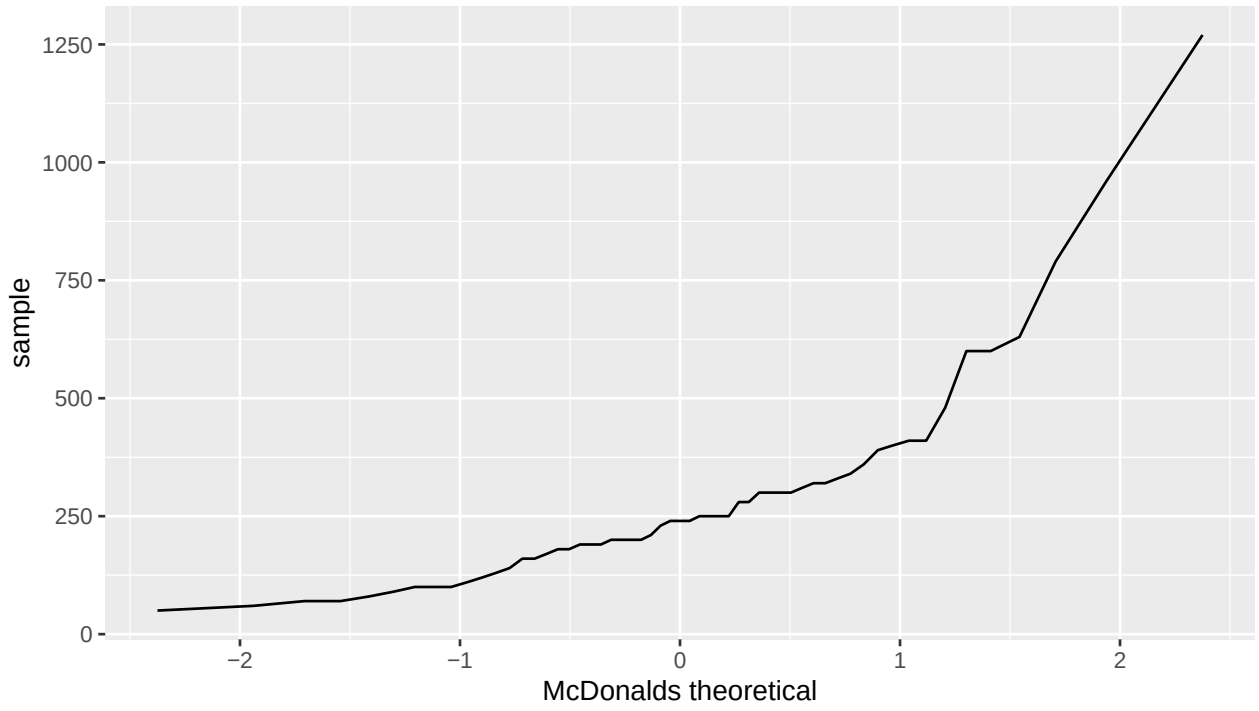


4. Does the normal probability plot for the calories from fat look similar to the plots created for the simulated data? That is, do the plots provide evidence that the female heights are nearly normal?

Answer: Yes, the normal probability plot for the calories from fat look similar to the plots created for the simulated data do look almost identical.

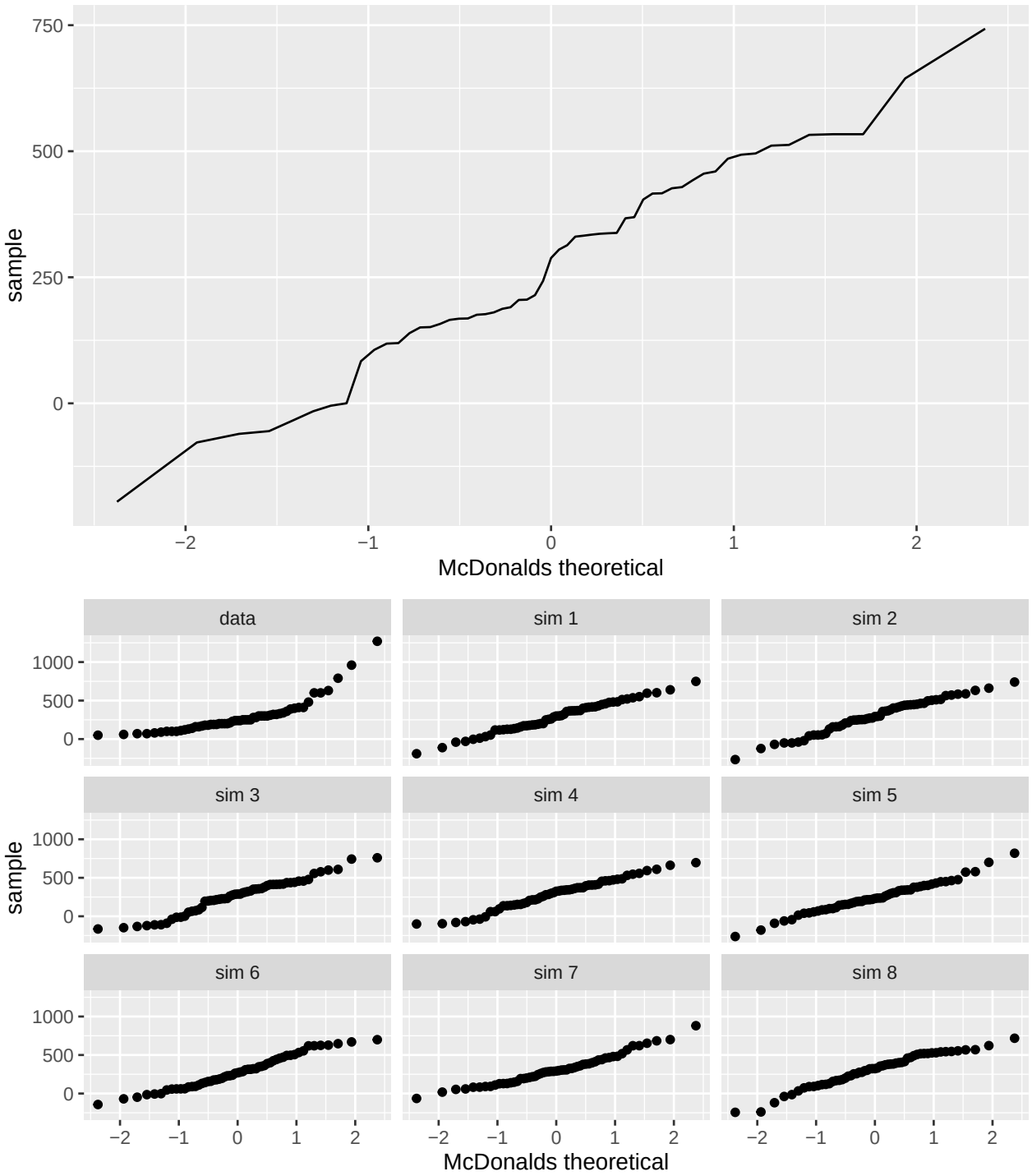
5. Using the same technique, determine whether or not the calories from McDonald's menu appear to come from a normal distribution.

Answer: Based the new plots, it is confirmed that McDonalds definitely is right skewed. The quantile-quantile and simulations also support that finding.




```
(sim_norm <- rnorm(n = nrow(mcdonalds), mean = mcd.mean, sd = mcd.sd))
```

```
##  [1]  485.1787460  644.4873670  742.8981739  -55.1698134  533.7054434
##  [6]  214.5389988  242.6438975   83.5714391  205.5706575  416.5154614
## [11]  139.1823139  337.9751417  176.8381266 -195.2318700  455.5358629
## [16]  332.4292220  168.3019446  404.1717212  493.0620898  205.0295787
## [21]  330.5779136  -77.6311712  495.6038928  180.3964320  511.0761334
## [26]  512.5490168  337.2418997  151.1820593  426.4542321   -4.7448507
## [31]  150.5331032  175.6515603  157.5066624  119.4876533  334.4973912
## [36]    0.1138098  106.0567829  118.4007447  167.8408691  -33.9071974
## [41]  -15.7846149  533.7086891  416.0252603  -60.6062106  532.4573924
## [46]  336.2838267  305.0360055  187.3759320  313.5168172  366.9892236
## [51]  369.1923042  288.2264682  428.8344535  459.8694518  165.5247458
## [56]  442.3211361  190.5690365
```



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 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?

6a. What is the probability that a randomly chosen Dairy Queen product has more than 40 grams of protein?

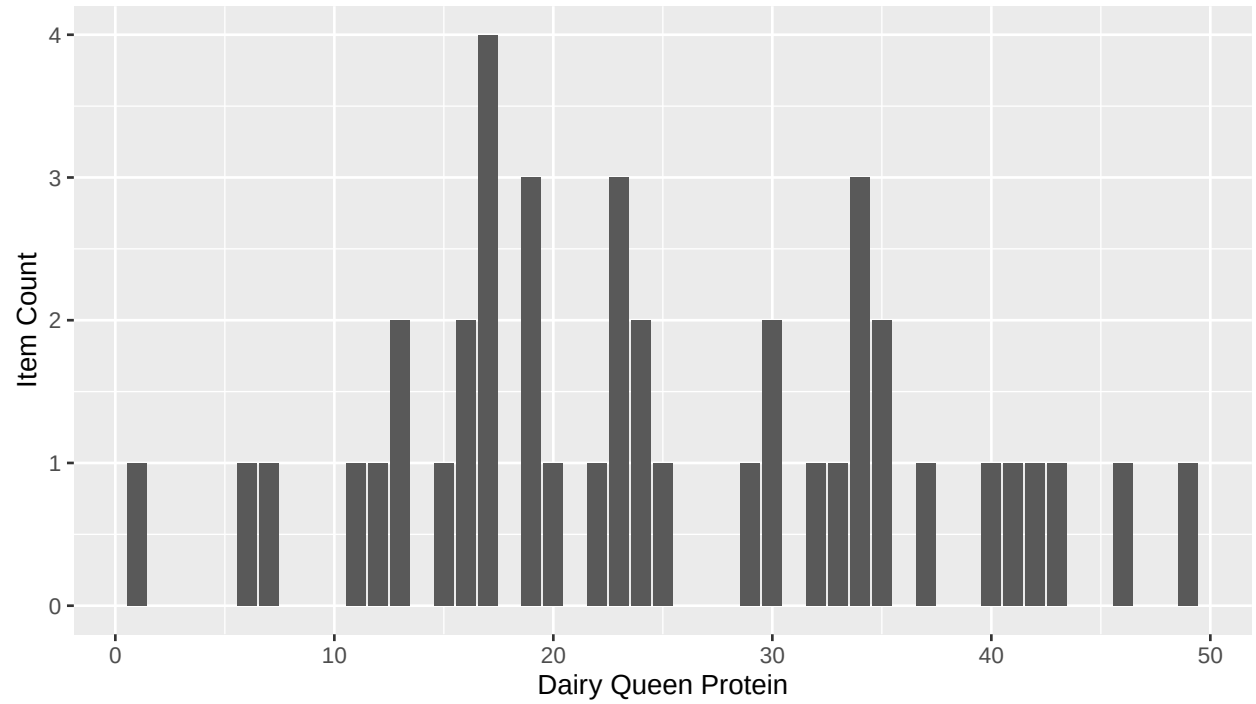
Answer: Caculate the mean and standard deviation of protein in DQ menu items;

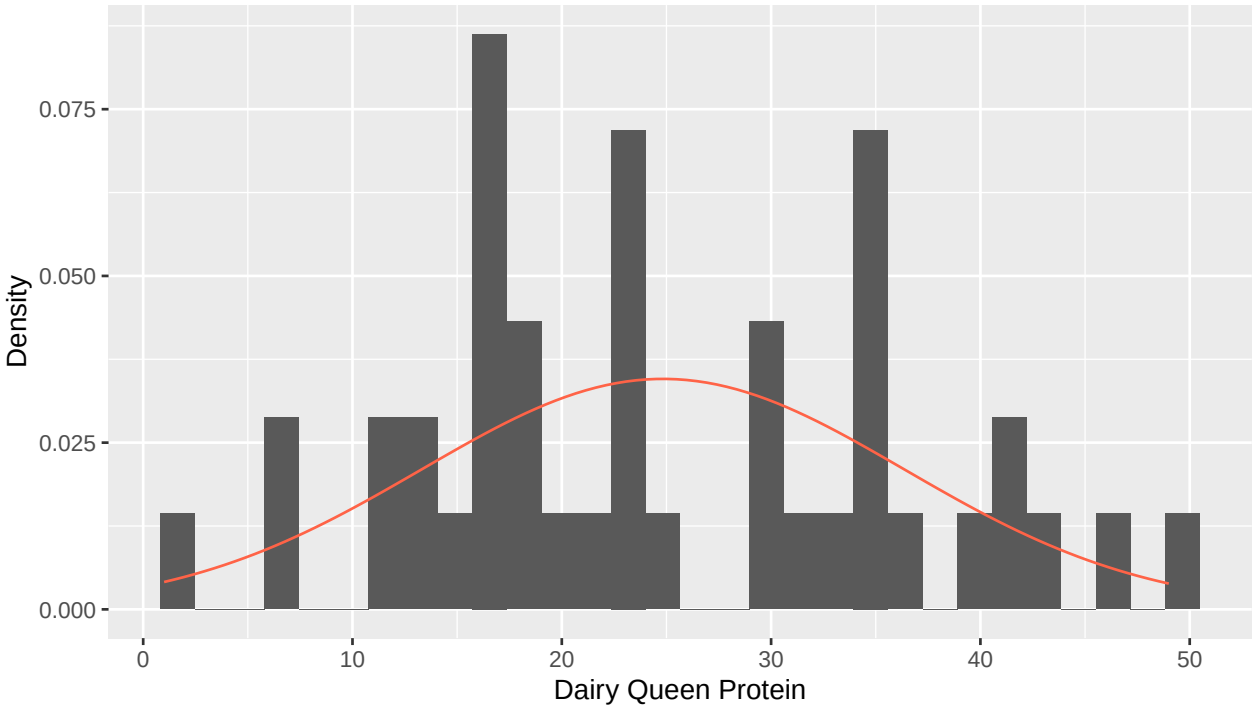
```
(dqpmean <- mean(dairy_queen$protein))
```

```
## [1] 24.83333
```

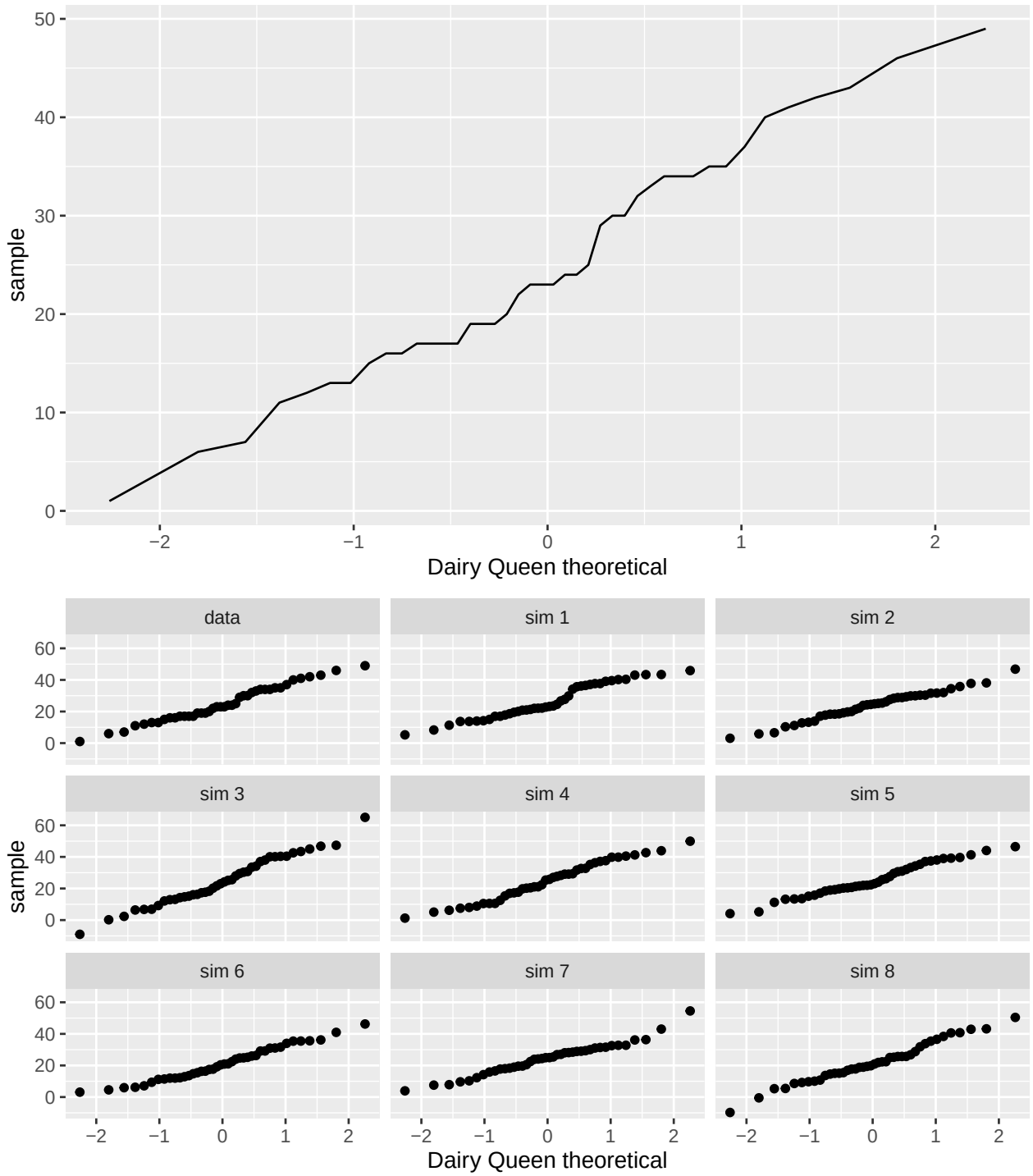
```
(dqpsd   <- sd(dairy_queen$protein))
```

```
## [1] 11.54401
```





Construct a normal probability Q-Q "quantile-quantile" plot for protein.



The normal probability plot for the grams of protein in a menu item look similar to the plots created for the simulated data very close to identical. This shows that the DQ distribution for protein content is somewhat close to normal.

Theoretically, there is a 9.5% probability that I can randomly select a menu item that has more than 40g of protein.

```
1 - pnorm(q = 40, mean = dqpmean, sd = dqpsd)
```

```
## [1] 0.09445468
```

Empirically, there is a 11.9% chance that I can randomly select a menu item that has more than 40g of proten.

```
dairy_queen %>%
  filter(protein > 40) %>%
  summarise(percent = n() / nrow(dairy_queen))
```

```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1   0.119
```

6b. What is the probability that a randomly chosen Dairy Queen product has more than 40 grams of carbs?

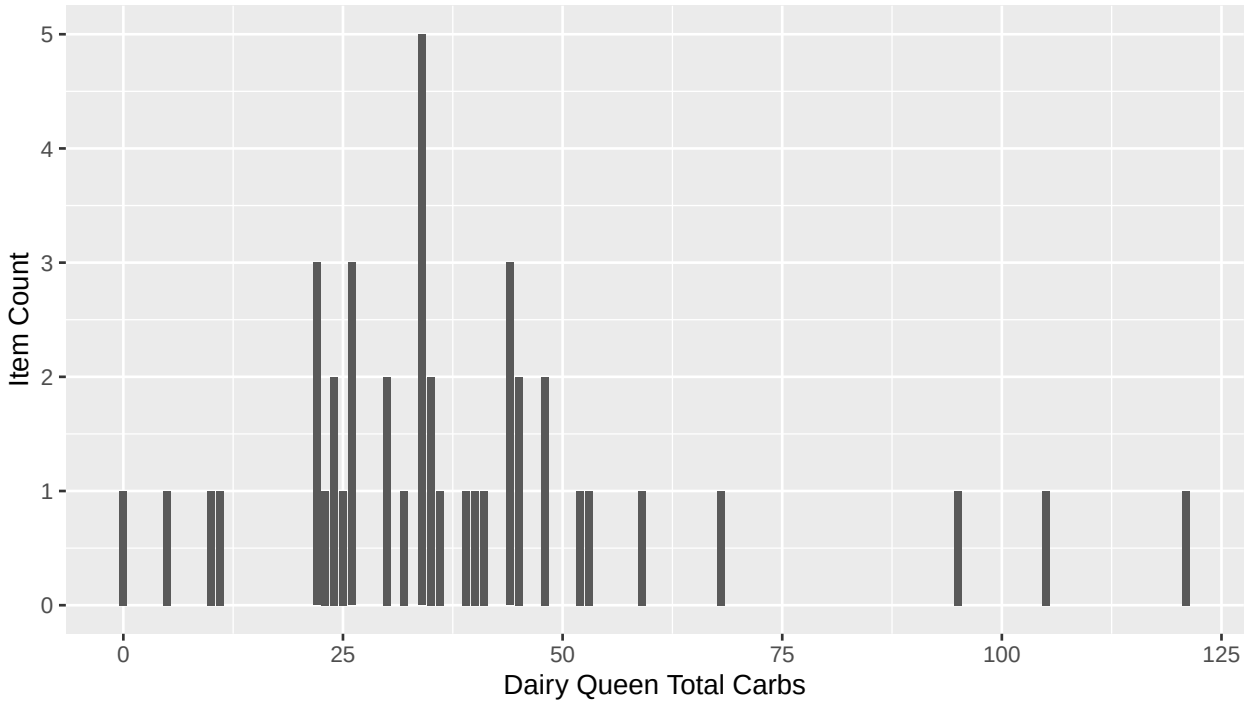
Answer: Calculate the mean and standard deviation of protein in DQ menu items;

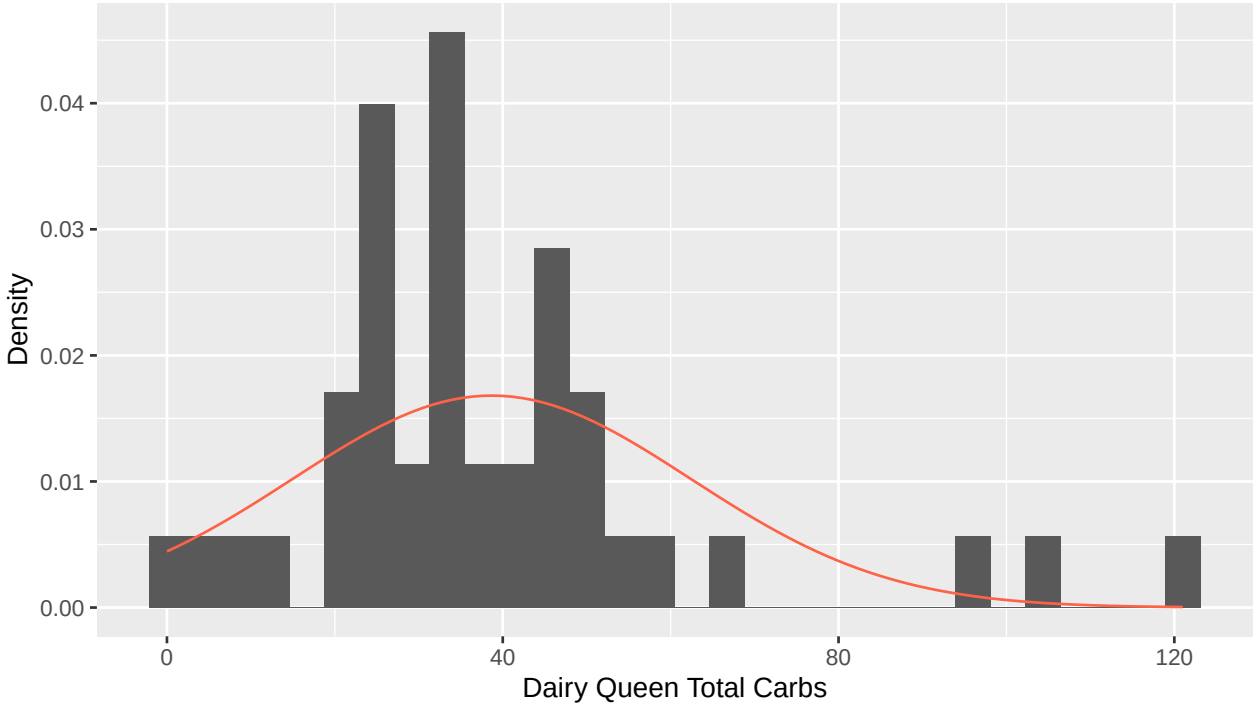
```
(dqcbmean <- mean(dairy_queen$total_carb))
```

```
## [1] 38.69048
```

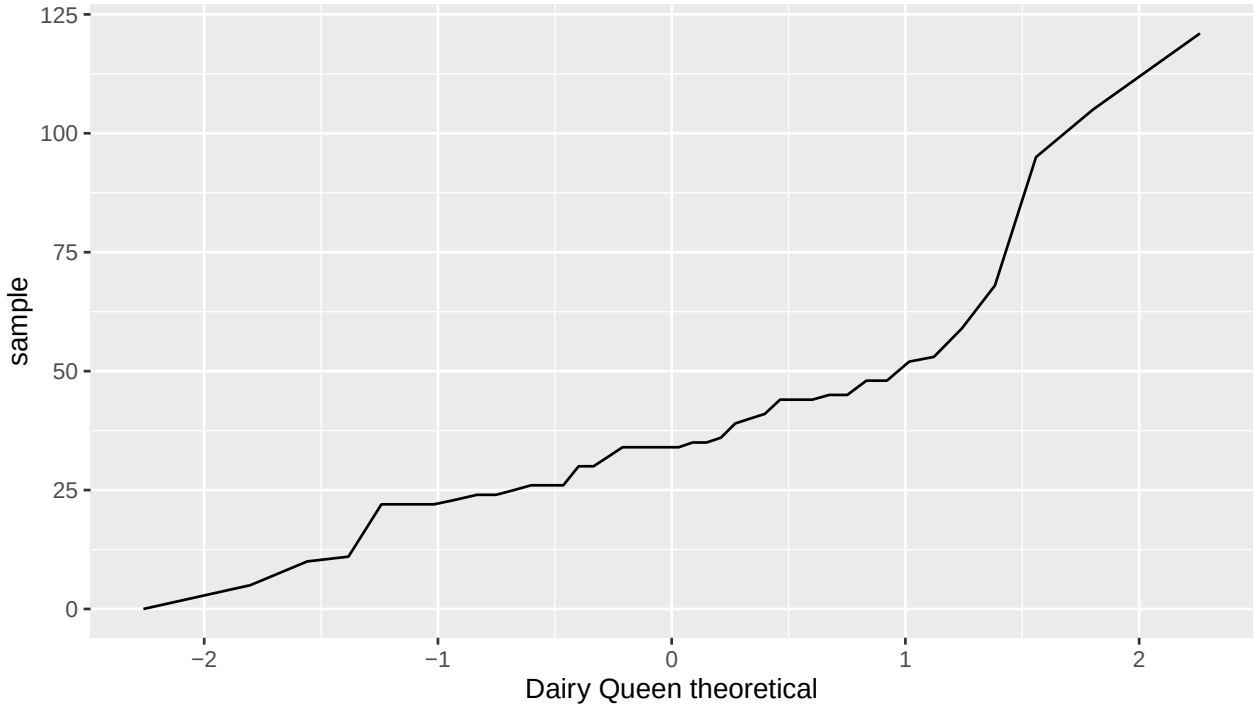
```
(dqcbsd   <- sd(dairy_queen$total_carb))
```

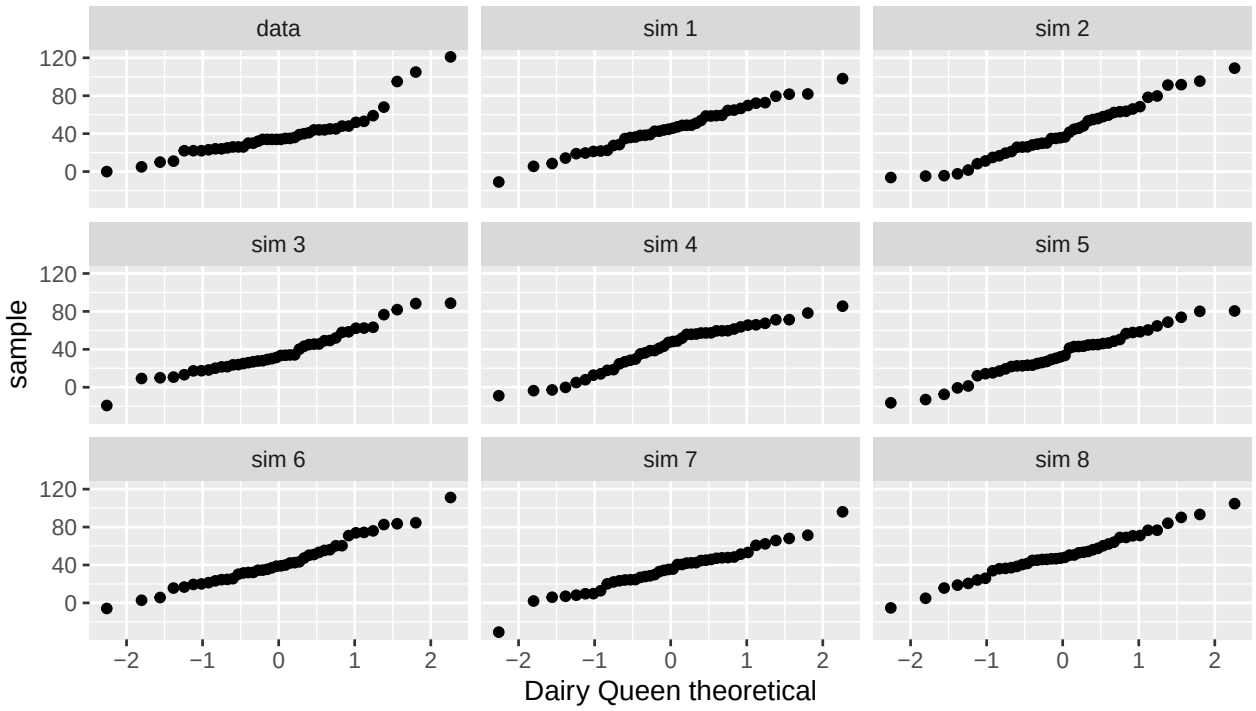
```
## [1] 23.72966
```





Construct a normal probability Q-Q "quantile-quantile" plot for total carbs





The normal probability plot for the grams of protein in a menu item look similar to the plots created for the simulated data very close to identical. This shows that the DQ distribution for protein content is somewhat close to normal.

Theoretically, there is a 39.5% probability that I can randomly select a menu item that has more than 45g total carbs.

```
1 - pnorm(q = 45, mean = dqcbmean, sd = dqcbsd)
```

```
## [1] 0.3951613
```

Empirically, there is a 21.43% chance that I can randomly select a menu item that has more than 40g of proten.

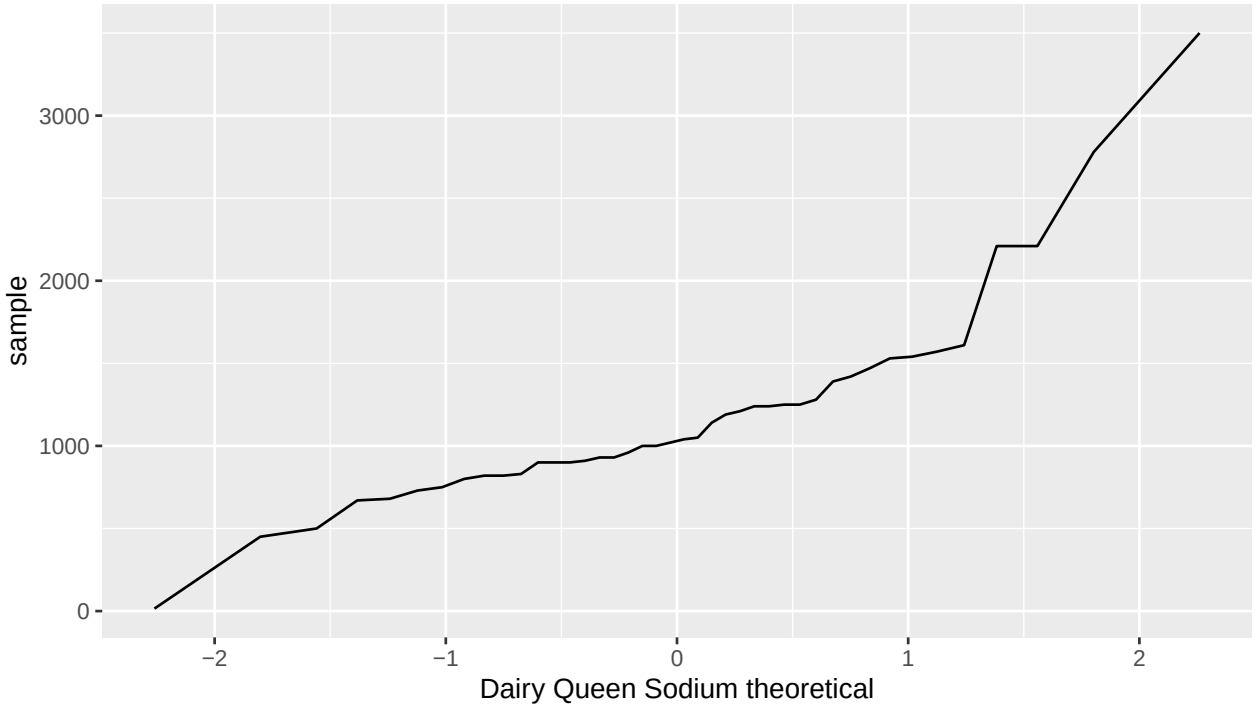
```
dairy_queen %>%
  filter(total_carb > 45) %>%
  summarise(percent = n() / nrow(dairy_queen))
```

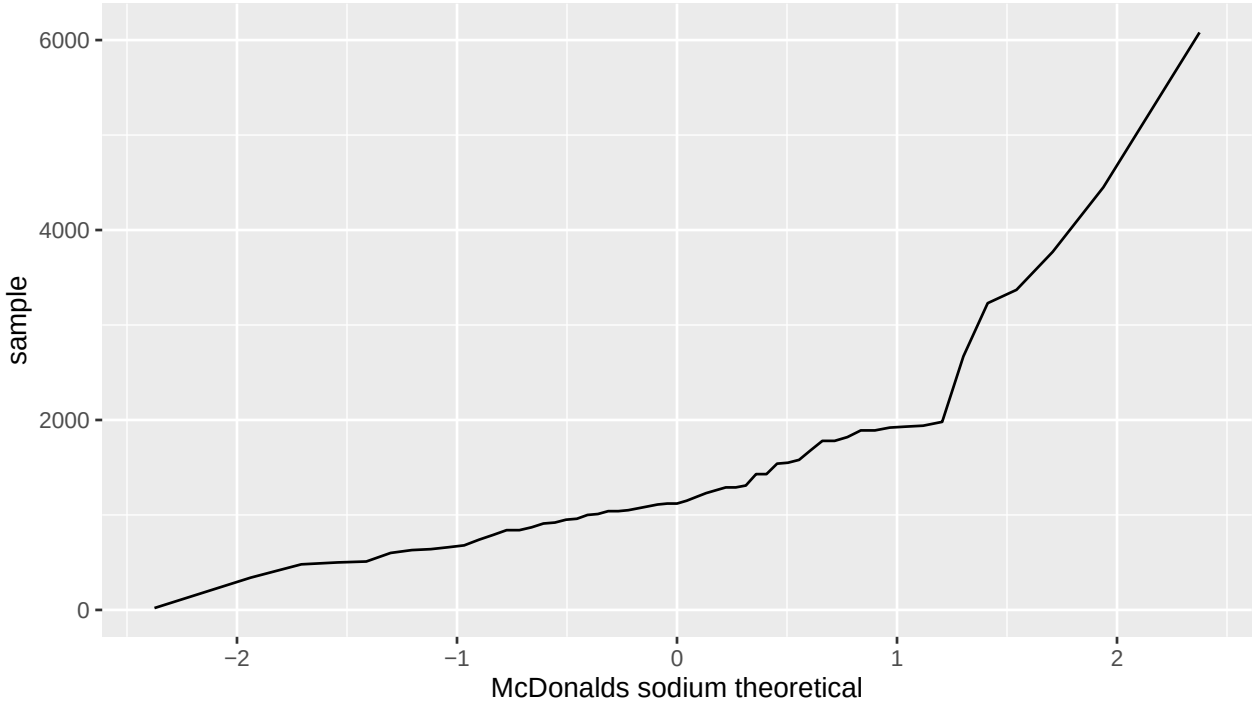
```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1   0.214
```

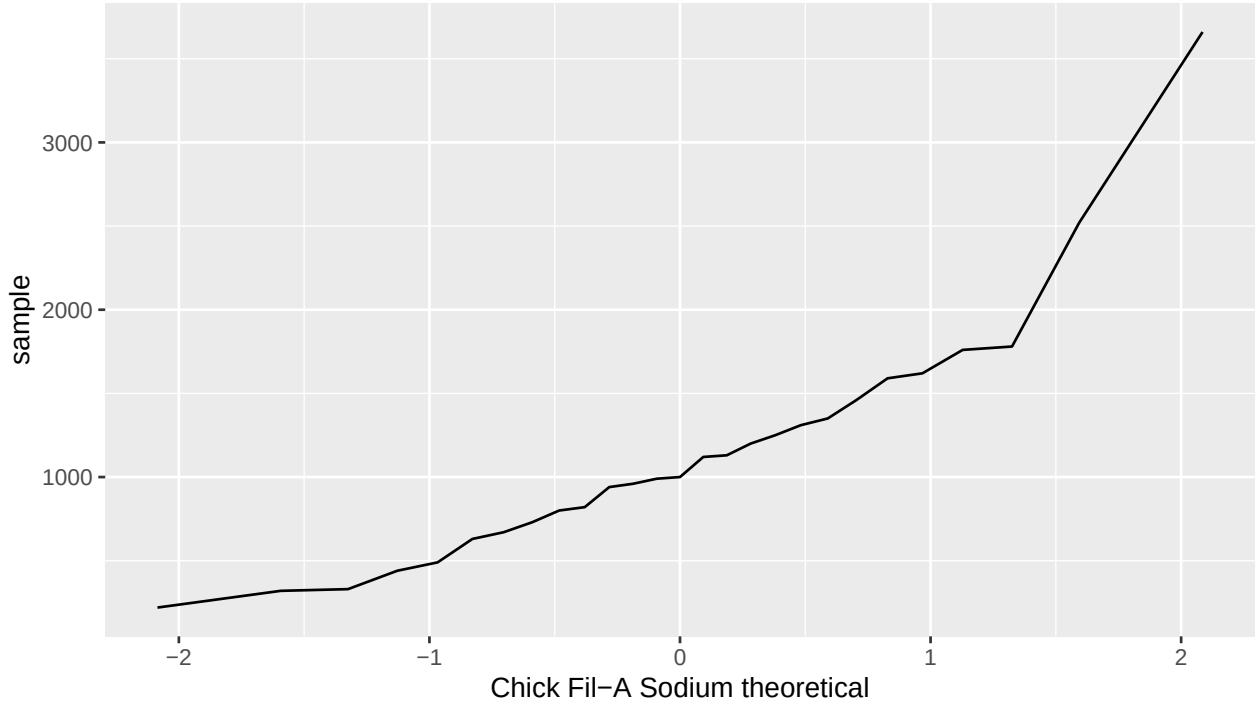
More Practice

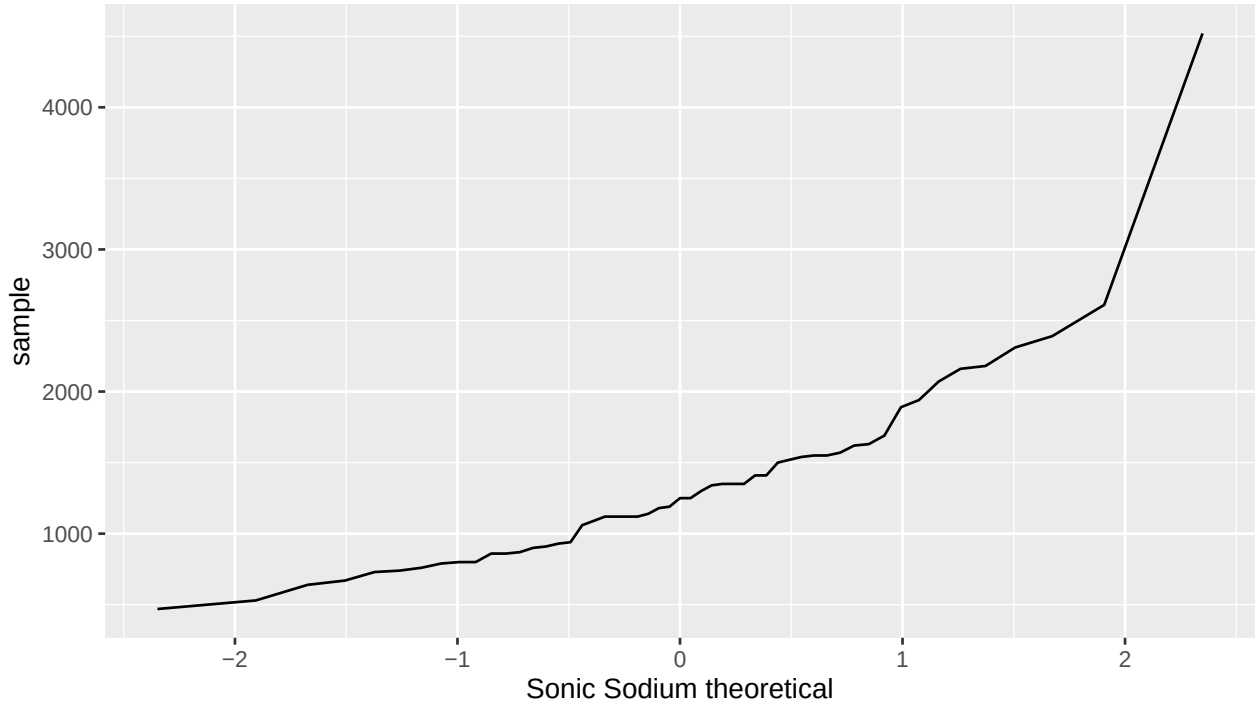
7. Now let's consider some of the other variables in the dataset. Out of all the different restaurants, which ones' distribution is the closest to normal for sodium?

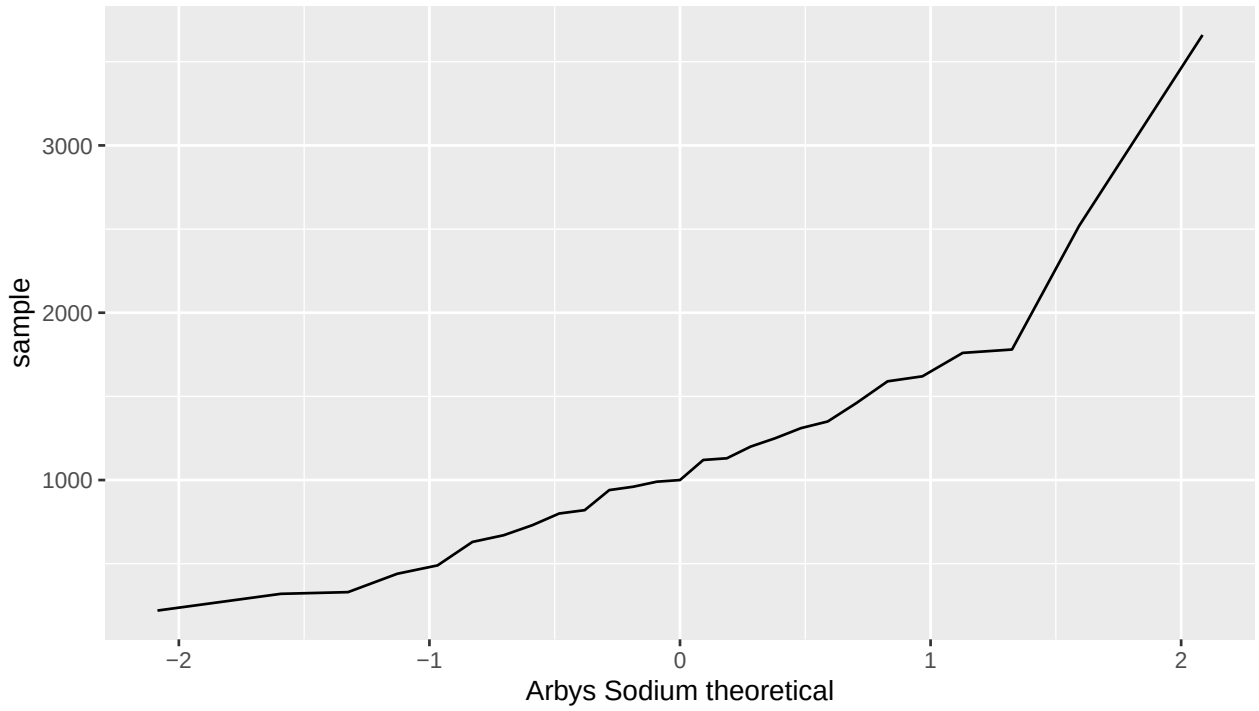
Answer: Burger King and Subway seem to have the closest distribution to normal for sodium of all the restaurants in the dataset.

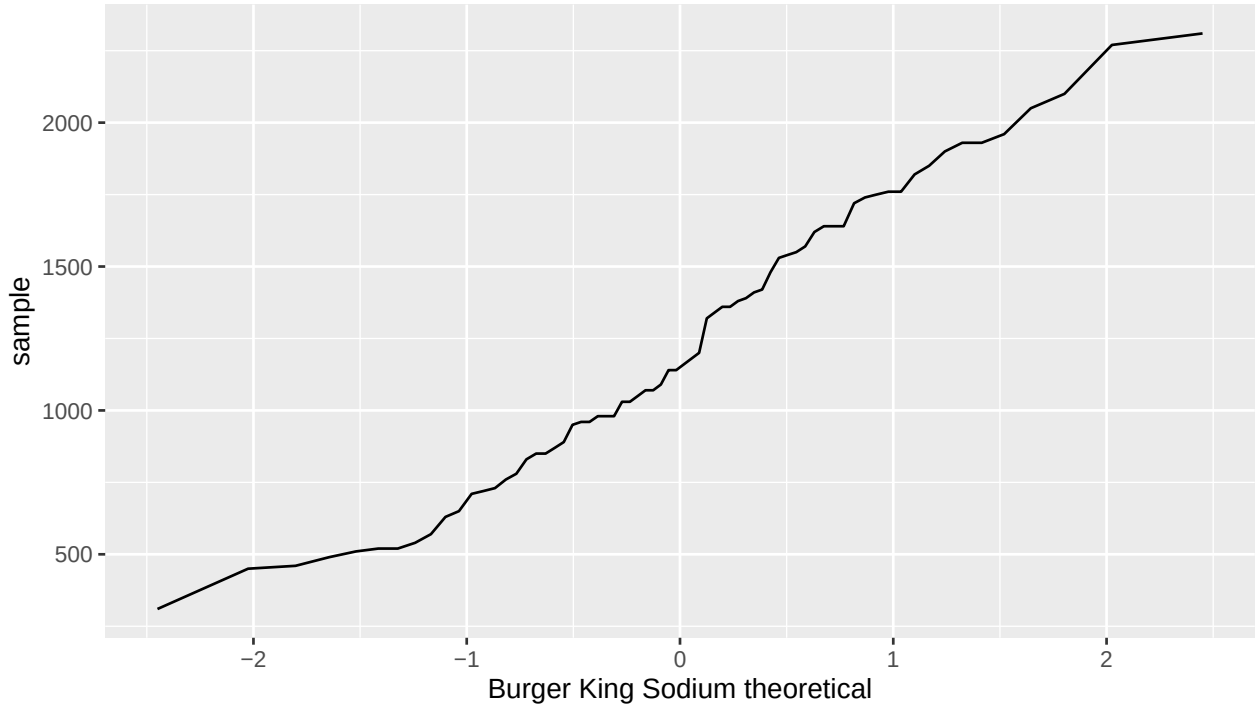


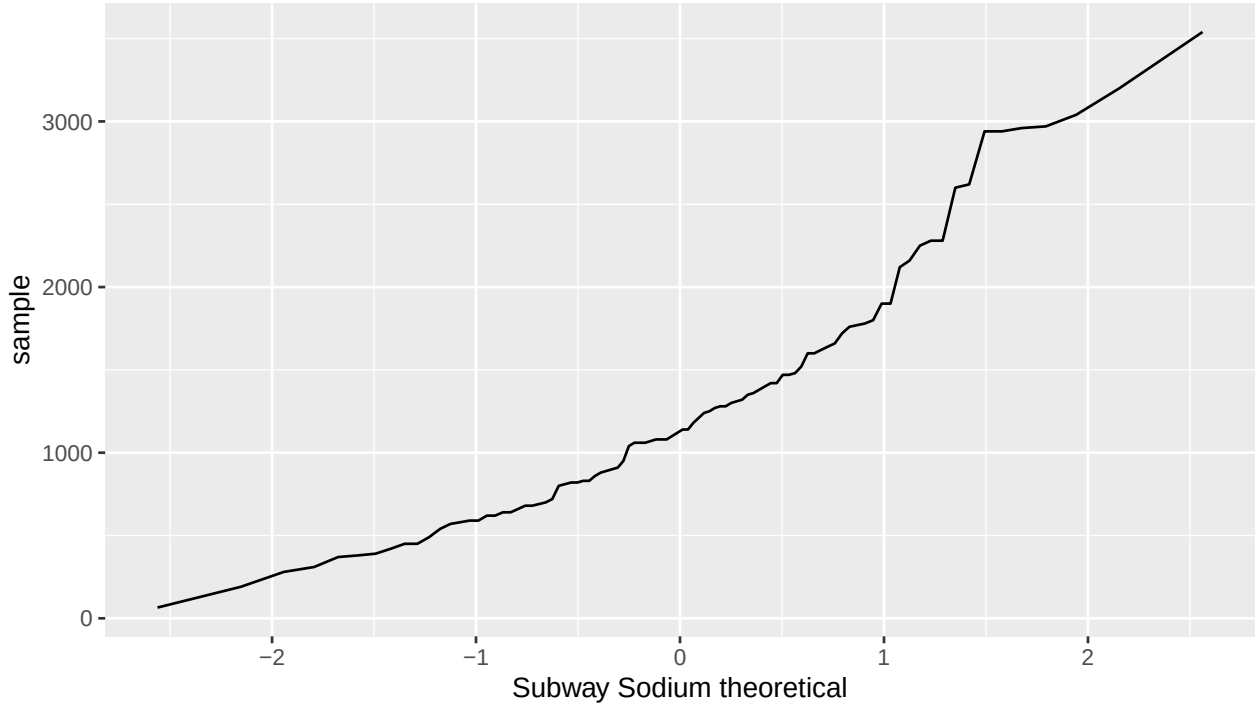


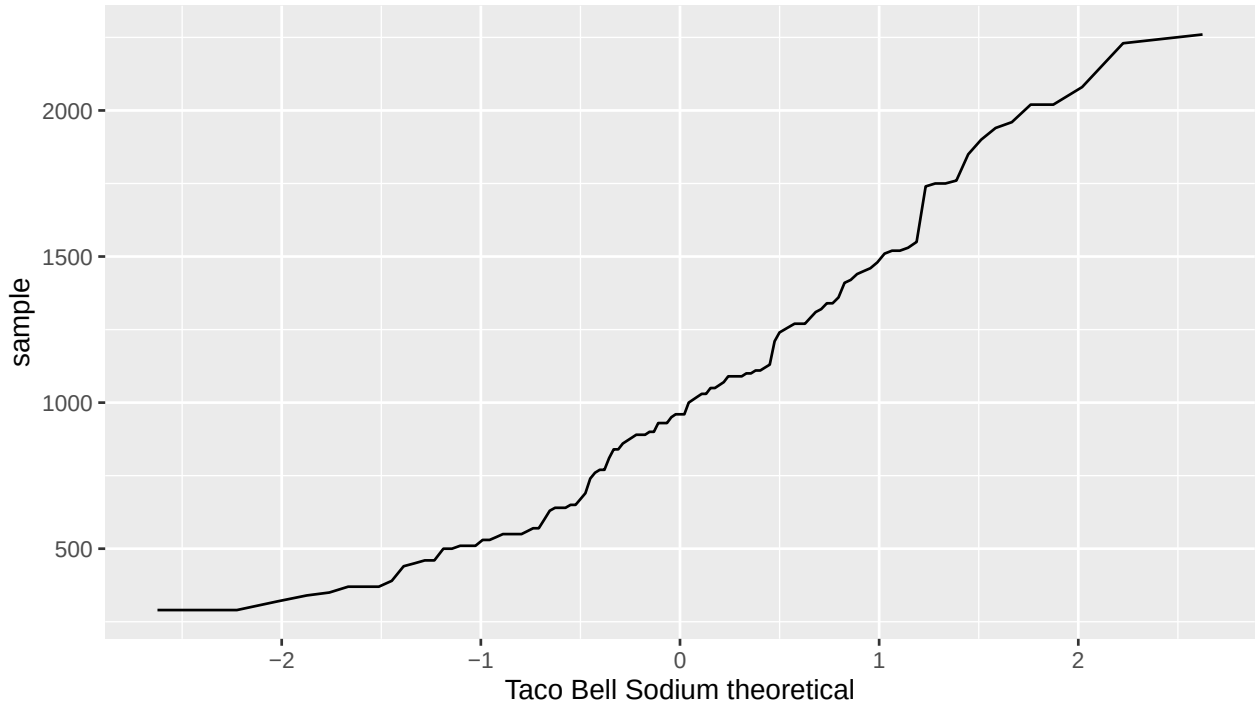

sample
3000
2000
1000
−2
−1
0
1
2
Chick Fil−A Sodium theoretical


sample
4000
3000
2000
1000
−2
−1
0
1
2
Sonic Sodium theoretical


3000
2000
1000
sample
−2
−1
0
1
2
Arbys Sodium theoretical


2000
1500
1000
500
sample
−2
−1
0
1
2
Burger King Sodium theoretical





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?

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.

Answer: See 6B. The histogram for total carbs for Dairy Queen is slightly right skewed.